Resource modalities in tensor logic

Paul-André Melliès, Nicolas Tabareau

Resource modalities in tensor logic

Paul-André Melliès   Nicolas Tabareau *

Abstract

The description of resources in game semantics has never achieved the simplicity and precision of linear logic, because of the misleading conception that linear logic is more primitive than game semantics. Here, we defend the opposite view, and thus advocate that game semantics is conceptually more primitive than linear logic. This revised point of view leads us to introduce tensor logic, a primitive variant of linear logic where negation is not involutive. After formulating its categorical semantics, we interpret tensor logic in a model based on Conway games equipped with a notion of payoff, in order to reflect the various resource policies of the logic: linear, affine, relevant or exponential.

Keywords: Game semantics, Conway games, linear logic, tensor logic, resource modalities, linear negation, categorical semantics.

1 Introduction

Game semantics and linear logic. Born (or rather reborn) at the beginning of the 1990s in the turmoil produced by the discovery of linear logic by Girard [13], game semantics remained under its spiritual influence for a very long time. This patronage was extraordinarily healthy and profitable in the early days: properly guided, game semantics developed steadily, along the idea that every *formula* of linear logic describes a *game*, and that every *proof* of the formula describes a *strategy* for playing that game. This correspondence between formulas of linear logic and games is supported by a series of elegant and striking analogies. One basic principle of linear logic is that negation

$$A \mapsto \neg A$$

is involutive. This means that every formula A is equal (or at least isomorphic) to the formula negated twice:

$$A \quad \cong \quad \neg\neg A. \tag{1}$$

This principle is nicely reflected in game semantics by the idea that negating a game A consists in permuting the roles of the two players. Hence, negating a game twice amounts

*This work is supported by the ANR Curry-Howard Correspondence and Concurrency Theory (CHOCO). Postal address: Laboratoire PPS, Université Paris Denis Diderot, 2 place Jussieu, Case 7014, 75251 Paris Cedex 05, FRANCE. Email addresses: **mellies@pps.jussieu.fr** and **tabareau@pps.jussieu.fr**

to permuting the role of Proponent and Opponent twice, which is just like doing nothing. Typically, if A is a chess board where White starts, $\neg A$ is the chess board where Black starts, and $\neg\neg A$ is again the chess board where White starts.

Another basic principle of linear logic is that every formula behaves as a resource, which disappears once consumed. In particular, a proof of the formula

$$A \multimap B$$

enables to deduce the conclusion B by using (one should say : consuming) its hypothesis A – seen here as a resource – exactly once. Again, this principle is nicely reflected in game semantics, by the idea that playing a game is just like consuming a resource: the game itself.

The connectives of linear logic are also nicely reflected in game semantics. For instance, the tensor product

$$A \otimes B$$

of two formulas A and B is suitably interpreted as the game (or formula) A played in parallel with the game (or formula) B, where only Opponent may switch from a component to the other one. This amounts to place two boards on the same table and to say that Black must respond on the board where White has just played. Similarly, the sum

$$A \oplus B$$

of two formulas A and B is nicely interpreted as the game where Proponent plays the first move, which consists in choosing between the game A and the game B, before carrying on in the selected component. This amounts to place two boards on the same table and to let Black decides if he wants to play on the left or right board. This choice is then irreversible. Finally, the exponential modality of linear logic

$$!A$$

applied to the formula A is interpreted as the game where several copies of the game A are played in parallel, and only Opponent is allowed (a) to switch from a copy to another and (b) to open a fresh copy of the game A. This amounts to play on parallel chess boards as for tensor but with the ability for White to add a new chess board to those already there.

What we describe here is in essence the game semantics of linear logic described by Blass [10]. The model is simple and elegant, and reflects the full flavor of the resource policies of linear logic. Interestingly, Blass described the linear decomposition of intuitionistic implication in a much earlier game semantics, which may be thus seen as a precursor of linear logic [9]. Another important precursor is the sequential algorithm model defined by Berry and Curien, which provided the first interactive model of the programming language PCF [8]. From an historical point of view, it should be also mentioned that game semantics was revisiting and extending ideas previously investigated by Lorenzen's school [27, 28].

A schism with linear logic. The destiny of game semantics has been to emancipate itself from linear logic in the mid–1990s, in order to comply with its own designs, inherited from denotational semantics:

1. the desire to interpret *programs* written in programming languages with effects (recursion, states, etc.) and to characterize exactly their interactive behavior inside *fully abstract* models;

2. the desire to understand the algebraic principles of programming languages and effects, using the language of category theory.

A new generation of game semantics arose, propelled by (at least) two different lines of research:

1. Abramsky and Jagadeesan [2] noticed that the (alternating variant of the) Blass model does not define a categorical model of linear logic. Worse: it does not even define a category, for lack of associativity. Abramsky calls this phenomenon the *Blass problem* and describes it in [1].

2. Hyland and Ong [18] introduced the notion of *arena game*, and characterized the interactive behavior of programs written in the functional language PCF — the simply-typed λ-calculus with conditional test, arithmetic and recursion. A similar result with a slightly different model based on the geometry of interaction has been obtained by Abramsky, Malacaria and Jagadeesan [3]. Note that despite their publication dates, those works have both been done during 1994.

So, the Blass problem indicates that it is difficult to construct a (sequential) game model of linear logic. At the same time, arena games became mainstream in the mid–1990s, although they do not define a model of linear logic. These two reasons (at least) opened a schism between game semantics and linear logic: it suddenly became accepted that categories of (sequential) games and strategies would only capture *fragments* of linear logic (intuitionistic or polarized) but not the whole thing.

A conciliation through tensor logic. In order to understand properly how the resource modalities of linear logic may be adapted to game semantics, it appears necessary to reunify the two subjects. Since the disagreement started with category theory, we believe that this reunification should occur at the categorical level. We explain (in §2) how to achieve this by *relaxing* the involutive negation of linear logic into a less constrained tensorial negation. This negation induces in turn a *linear continuation* monad, whose unit

$$\eta_A \quad : \quad A \quad \longrightarrow \quad \neg\neg A \tag{2}$$

refines the isomorphism (1) of linear logic. Moving from an involutive to a tensorial negation means that we replace linear logic by a more general and primitive logic – which we call *tensor logic*. As we will see, this shift to tensor logic clarifies the Blass problem, and describes the structure of arena games. It also enables to express resource modalities in game semantics, just as it is usually done in linear logic.

Tensor logic provides a new insight on polarization in logic, an idea discovered by Girard in his work on classical logic and system LC [14] and later exploited by Laurent in his work on polarized linear logic [25]. Indeed, an unexpected phenomenon shows up in these polarized logic: the resource modality $!_{\mathrm{pol}}$ changes the polarity of formulas. This peculiar fact is nicely explained in tensor logic, by the decomposition of the modality $!_{\mathrm{pol}}$ into two constructors: the exponential modality ! which does not change the polarity of formulas, and the tensorial negation, noted $\downarrow$ in this framework (rather that $\neg$), whose role as a negation is to swap the point of view of Opponent and Proponent on a formula, which amounts to reversing its polarity:

$$\underset{\mathrm{pol}}{!}\ A \quad = \quad !\ \downarrow\ A.$$

More generally, we would like to promote a radical change of perspective on polarization. As we see it, tensor logic is not reduced to a fragment of linear logic, as one generally thinks of polarized logic. On the contrary, we defend the thesis that tensor logic is a more primitive logic that linear logic, closer to the mechanisms of continuations described by game semantics. And in the same way that classical logic is interpreted in intuitionistic logic through the Gödel translation, we will see that linear logic is interpreted in tensor logic through a similar translation, of a categorical nature: namely, a Kleisli construction. In a word, tensor logic is to linear logic what intuitionistic logic is to classical logic: a formalism closer to computations and programs. This guiding principle that linear logic should be seen as a "depolarized tensor logic" emerged from the semantic work of the first author on asynchronous games, where the game model of polarized logic was "depolarized" and extended in this way to full propositional linear logic [31, 32].

Plan of the paper. We describe (§2) a categorical semantics of resources in game semantics, and explain (§3) in what sense the resulting topography refines both linear logic and polarized logic. After that, we construct (§4) a compact-closed (that is, self-dual) category inspired by Conway games, where the resource policy is enforced by a notion of payoff. From this, we derive (§5) a model of our categorical semantics of tensor logic with resource modalities, using a family construction, and conclude (§6).

2 Categorical models of resources

Despite the extraordinary efficiency of game semantics to interpret fragments of linear logic, serious difficulties arise when one tries to go beyond these fragmentary models, in order to construct a (sequential) game model of full propositional linear logic. One main technical obstacle was described by Abramsky and Jagadeesan as the Blass problem [2]. This says that the sequential game semantics described by Blass [10] does not give rise to a category, because the expected definition of composition between strategies is not associative. This alarming situation convinced the first author to develop a theory of asynchronous games in order to resolve the Blass problem, and to construct a game-theoretic model of full propositional linear logic [30, 31, 32]. Curiously, the solution requires to consider sequential strategies modulo a quotient – a quotient which, in the light of a categorical observation of Hasegawa [16] amounts to identifying the two canonical morphisms

$$\neg\neg A \ \otimes\ \neg\neg B \quad \rightrightarrows \quad \neg\neg\ (A \otimes B)$$

induced by the strength of the continuation monad. The point is that identifying the two strategies makes the continuation monad T a commutative monad – this implying in turn that the full subcategory of negated objects (thus of the form $\neg A$) defines a $*$-autonomous category, and thus a model of linear logic.

Since the construction of the game-theoretic model of linear logic requires to identify strategies modulo a notion of quotient, it is natural to step back, and to investigate the status of a logic of tensor and negation where the continuation monad would not be commutative. This idea leads to tensor logic – a more primitive logic deeply related to game semantics. It appears then that a major part of the models of linear logic are constructed in this way, starting from a model of tensor logic where the continuation monad is commutative. This is typically the case for phase spaces, coherence spaces, or finiteness spaces [12]. One interesting aspect of our approach is that one can define *different* negations on the same model of tensor logic – this inducing different but harmoniously related models of linear logic. See the PhD thesis of the second author for a combined study of coherence spaces and finiteness spaces [42].

We introduce now the notion of *dialogue category* which is a symmetric monoidal category equipped with a *tensorial negation*. We then explain how such a dialogue category may be equipped with coproducts and various resource modalities. The first author describes in [33] how to extract a syntax of proofs from a categorical semantics, by using string diagrams and functorial boxes. In this case, we call *tensor logic* the resulting logic of tensor and negation. We provide in Section 3 a sequent calculus for the logic, in order to compare it to linear logic and to polarized logic.

Tensorial negation. A *tensorial negation* on a symmetric monoidal category $(\mathcal{C}, \otimes, 1)$ is defined as a functor

$$\neg \quad : \quad \mathcal{C} \quad \longrightarrow \quad \mathcal{C}^{op} \tag{3}$$

together with a family of bijections

$$\varphi_{A,B,C} \quad : \quad \mathcal{C}(A \otimes B, \neg C) \quad \cong \quad \mathcal{C}(A, \neg(B \otimes C))$$

natural in A, B and C such that the diagram

$$\begin{tikzcd}
\mathcal{C}(A \otimes (B \otimes C), \neg\, D) \arrow[rr, "\mathcal{C}(\alpha_{A,B,C}, \neg C)"] \arrow[dd, "\varphi_{A,B\otimes C,D}"'] & & \mathcal{C}((A \otimes B) \otimes C, \neg\, D) \arrow[d, "\varphi_{A\otimes B,C,D}"] \\
& & \mathcal{C}(A \otimes B, \neg\,(C \otimes D)) \arrow[d, "\varphi_{A,B,C\otimes D}"] \\
\mathcal{C}(A, \neg\,((B \otimes C) \otimes D)) \arrow[rr, "\mathcal{C}(A, \neg \alpha^{-1}_{B,C,D})"'] & & \mathcal{C}(A, \neg\,(B \otimes (C \otimes D)))
\end{tikzcd}$$

commutes. A symmetric monoidal category equipped with a tensorial negation is called a *dialogue category*. Given a tensorial negation, it is customary to define falsity as the object $\bot = \neg\, 1$ obtained by "negating" the unit object 1 of the monoidal category. Note that we use the notation 1 (instead of I or e) in order to remain consistent with the notations of linear logic. Note also that the bijection $\varphi_{A,B,1}$ defines a one-to-one correspondence

$$\varphi_{A,B,1} \quad : \quad \mathcal{C}(A \otimes B, \bot) \quad \cong \quad \mathcal{C}(A, \neg B)$$

for all objects A and B. For that reason, the definition of a negation $\neg$ is equivalent to the (somewhat informal) statement that "the object $\bot$ is exponentiable" in the symmetric monoidal category $\mathcal{C}$, with negation $\neg A$ noted $\bot^A$. More on this topic will be found in the survey [34].

Self-adjunction. One fundamental aspect of the notion of tensorial negation is that negation seen as a functor (3) is left adjoint to the opposite functor

$$\neg \quad : \quad \mathcal{C}^{op} \quad \longrightarrow \quad \mathcal{C} \tag{4}$$

This comes from the natural bijection

$$\mathcal{C}(A, \neg B) \quad \cong \quad \mathcal{C}(B, \neg A) \quad = \quad \mathcal{C}^{op}(\neg A, B).$$

This fundamental "self-adjunction" phenomenon was already mentioned by Kock [23] and was then rediscovered by Thielecke [43] in his PhD thesis. This observation plays a key role in an unpublished work by Selinger and the first author on polar categories, a categorical semantics of polarized logic, continuations and arena games [35]. More generally, the idea of adjunction also appears in the study of games and distributors (seen as polarized categories) by Cockett and Seely [11].

Continuation monad. Every tensorial negation $\neg$ induces a self-adjunction, and thus a monad

$$\neg\neg \quad : \quad \mathcal{C} \longrightarrow \mathcal{C}.$$

This monad is called the *continuation monad* of the negation. One fundamental fact observed by Moggi [39] is that the continuation monad is *strong* but not commutative in general. By strong monad, we mean that the monad $\neg\neg$ is equipped with a family of morphisms:

$$t_{A,B} \quad : \quad A \otimes \neg\neg B \quad \longrightarrow \quad \neg\neg\,(A \otimes B)$$

natural in A and B, and satisfying a series of coherence properties. By commutative monad, we mean a strong monad making the two canonical morphisms

$$\neg\neg A \otimes \neg\neg B \quad \rightrightarrows \quad \neg\neg\,(A \otimes B) \tag{5}$$

coincide. A tensorial negation $\neg$ is called *commutative* when the continuation monad induced in $\mathcal{C}$ is commutative — or equivalently, a monoidal monad in the lax sense.

Linear implication. A dialogue category $\mathcal{C}$, with negation $\neg$ is not very far from being monoidal *closed*. It is possible indeed to define a *linear implication* $\multimap$ when its target $\neg B$ is a negated object:

$$A \multimap \neg B \quad \stackrel{\text{def}}{=} \quad \neg\,(A \otimes B).$$

In this way, the functor (4) defines what we call an *exponential ideal* in the category $\mathcal{C}$. When the functor is faithful on objects and morphisms, we may identify this exponential ideal with the subcategory of *negated objects* in the category $\mathcal{C}$. The exponential ideal discussed in McCusker's PhD thesis [29] arises precisely in this way.

Continuation category. Every dialogue category $\mathcal{C}$, with negation $\neg$, induces a *category of continuations* $\mathcal{C}^{\neg}$ with the same objects as $\mathcal{C}$, and morphisms defined as

$$\mathcal{C}^{\neg}(A, B) \quad \stackrel{\text{def}}{=} \quad \mathcal{C}(\neg A, \neg B).$$

Note that the category $\mathcal{C}^{\neg}$ is the opposite of the Kleisli category associated to the continuation monad in $\mathcal{C}$. Alternatively, the category $\mathcal{C}^{\neg}$ may be seen as the Kleisli category associated to the comonad in $\mathcal{C}^{op}$ induced by the adjunction.

Semantics of resources. A *resource modality* on a symmetric monoidal category $(\mathcal{C}, \otimes, e)$ is defined as an adjunction:

$$\begin{tikzcd} \mathcal{M} \arrow[r, bend left, "U"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "F"] \end{tikzcd} \tag{6}$$

where

- $(\mathcal{M}, \bullet, u)$ is a symmetric monoidal category,
- U is a symmetric monoidal functor.

Recall that a *symmetric monoidal* functor U is a functor which transports the symmetric monoidal structure of $(\mathcal{M}, \bullet, u)$ to the symmetric monoidal structure of $(\mathcal{C}, \otimes, e)$, up to isomorphisms satisfying suitable coherence properties. Another more conceptual definition of a resource modality is possible: it is an adjunction defined in the 2-category of symmetric monoidal categories, *lax* symmetric monoidal functors, and monoidal transformations. Now, the resource modality is called

- *affine* when the unit u is the terminal object of the category $\mathcal{M}$,
- *relevant* when every object of $\mathcal{M}$ is duplicable, that is when there exists a diagonal

  $$\delta_A \quad : \quad A \longrightarrow A \otimes A$$

  natural in A, compatible with the symmetry and satisfying the associativity diagram

$$\begin{tikzcd} A \arrow[r, "\delta_A"] \arrow[dr, "\delta_A"'] & A \otimes A \arrow[d, "symm_{A,A}"] \\ & A \otimes A \end{tikzcd} \tag{7} \qquad \begin{tikzcd} A \arrow[r, "\delta_A"] \arrow[d, "\delta_A"'] & A \otimes A \arrow[d, "A \otimes \delta_A"] \\ A \otimes A \arrow[r, "\delta_A \otimes A"'] & A \otimes A \otimes A \end{tikzcd} \tag{8}$$

- *exponential* when the tensor product $\bullet$ is a cartesian product, and the unit u is the terminal object of the category $\mathcal{M}$.

This definition of resource modality is inspired by the categorical semantics of linear logic, and more specifically by Benton's notion of Linear-Non-Linear model [7] — which may be now reformulated as a symmetric monoidal *closed* category $\mathcal{C}$ equipped with an exponential modality in our sense. Very often, we will identify the resource modality and the induced comonad $! \; = \; U \circ F$ on the category $\mathcal{C}$. We sum up the different resource modalities in the following table.

Modality	Category $(M, \otimes, e)$
Affine	the unit e is terminal
Relevant	every object is duplicable
Exponential	the structure is cartesian

The work of Jacobs on affine and relevant modalities [20] is based on a commutative monad on a cartesian closed category. He then considers the Eilenberg-Moore category induced by this (affine or relevant) monad in order to construct models of intuitionistic linear logic equipped with an affine or relevant modality. In particular, his construction is limited to the special case of models of intuitionistic linear logic obtained as categories of algebras.

3 Tensor logic

In our algebraic philosophy, tensor logic is entirely described by its categorical semantics, which is defined in the following way. First of all, every dialogue category $\mathcal{C}$ defines a model of *multiplicative* tensor logic. It then defines a model of *multiplicative additive* tensor logic when the category $\mathcal{C}$ has finite coproducts (noted $\oplus$) which *distribute* over the tensor product: this means that the canonical morphisms

$$(A \otimes B) \oplus (A \otimes C) \quad \longrightarrow \quad A \otimes (B \oplus C)$$

$$0 \quad \longrightarrow \quad A \otimes 0$$

are isomorphisms. Then, a model of full propositional tensor logic is defined as a model of multiplicative additive tensor logic, equipped with an exponential resource modality (with comonad noted $\underset{e}{!}$) as well as, ideally, an affine resource modality (with comonad noted $\underset{w}{!}$) and a relevant resource modality (with comonad noted $\underset{c}{!}$). A diagrammatic syntax of tensor logic may be then extracted from its categorical definition, along the line of [33]. However, we find useful to give a more familiar presentation of tensor logic, in order to compare it to linear logic and to polarized linear logic. To that purpose, we formulate below the sequent calculus of tensor logic in two different but equivalent ways: either two-sided or one-sided.

Two-sided presentation. The formulas $A, B, \ldots$ of tensor logic (in its two-sided presentation) are constructed as follows:

multiplicatives	$1 \mid \neg A \mid A \otimes B$
additives	$0 \mid A \oplus B$
resource modalities	$\underset{w}{!}A \mid \underset{c}{!}A \mid \underset{e}{!}A$

The sequents are of two forms: $\Gamma \vdash A$ where Γ is a context, and A is a formula; $\Gamma \vdash$ where Γ is a context (the notation $[A]$ expresses the unessential presence of A in the sequent). The sequent calculus of the multiplicative fragment appears in Figure 1. The first four rules express the monoidal structure on $\mathcal{C}$, the two below define a tensorial negation and the last two just represent identity and composition of our category $\mathcal{C}$. Figure 2 describes the rules managing finite coproducts. Figure 3 depicts the expected rules for the exponential modality (those are the rules of the ! of linear logic). The rules for the affine modality, given in Figure 4, are the same as for the exponential modality, but without contraction. The rules for the relevant modality, given in Figure 5, are the same as for the exponential modality, but without weakening.

One-sided presentation. In order to switch to the one-sided formulation of tensor logic, we need to introduce polarities. The formulas that were on the right in the two-sided presentation remain there, and are called *positive*. Dually, the formulas on the left move on the right, and are now called *negative*.

Two-sided presentation		One-sided presentation
$\Gamma \vdash$	$\rightsquigarrow$	$\vdash \Gamma^*$
$\Gamma \vdash A$	$\rightsquigarrow$	$\vdash \Gamma^*, A$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \text{Tensor-Right} \qquad \frac{\Gamma_1, A, B, \Gamma_2 \vdash [C]}{\Gamma_1, A \otimes B, \Gamma_2 \vdash [C]}\ \text{Tensor-Left}$$

$$\frac{}{\vdash 1}\ \text{Unit-Right} \qquad \frac{\Gamma \vdash [A]}{\Gamma, 1 \vdash [A]}\ \text{Unit-Left}$$

$$\frac{\Gamma, A \vdash}{\Gamma \vdash \ \neg A}\ \text{Negation-Right} \qquad \frac{\Gamma \vdash A}{\Gamma, \neg A \vdash}\ \text{Negation-Left}$$

$$\frac{}{A \vdash A}\ \text{Axiom} \qquad \frac{\Gamma \vdash A \qquad A, \Delta \vdash [B]}{\Gamma, \Delta \vdash [B]}\ \text{Cut}$$

Figure 1. Multiplicative tensor logic: two-sided presentation

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}\ \text{Sum-Right-1}$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}\ \text{Sum-Right-2}$$

$$\frac{\Gamma, A \vdash [C] \qquad \Gamma, B \vdash [C]}{\Gamma, A \oplus B \vdash [C]}\ \text{Sum-Left}$$

No right introduction rule for the zero

$$\frac{}{\Gamma, 0 \vdash [A]}\ \text{Zero-Left}$$

Figure 2. Additive tensor logic: two-sided presentation

$$\frac{!_e\Gamma \vdash A}{!_e\Gamma \vdash \ !_e A}\ \text{Strengthening} \qquad \frac{\Gamma, A \vdash [B]}{\Gamma, !_e A \vdash [B]}\ \text{Dereliction}$$

$$\frac{\Gamma \vdash [B]}{\Gamma, !_e A \vdash [B]}\ \text{Weakening} \qquad \frac{\Gamma, !_e A, !_e A \vdash [B]}{\Gamma, !_e A \vdash [B]}\ \text{Contraction}$$

Figure 3. Exponential modality: two-sided presentation

$$\frac{!_w\Gamma \vdash A}{!_w\Gamma \vdash \ !_w A}\ \text{Strengthening} \qquad \frac{\Gamma, A \vdash [B]}{\Gamma, !_w A \vdash [B]}\ \text{Dereliction}$$

$$\frac{\Gamma \vdash [B]}{\Gamma, !_w A \vdash [B]}\ \text{Weakening}$$

Figure 4. Affine modality: two-sided presentation

$$\frac{!_c\Gamma \vdash A}{!_c\Gamma \vdash \ !_c A}\ \text{Strengthening} \qquad \frac{\Gamma, A \vdash [B]}{\Gamma, !_c A \vdash [B]}\ \text{Dereliction}$$

$$\frac{\Gamma, !_c A, !_c A \vdash [B]}{\Gamma, !_c A \vdash [B]}\ \text{Contraction}$$

Figure 5. Relevant modality: two-sided presentation

So, there are two kinds of sequents in this formulation: the sequents $\vdash \Gamma$ where Γ contains only negative formulas, and the sequents $\vdash \Gamma, P$ containing exactly one positive formula P, (the notation $[P]$ expresses the unessential presence of P in the sequent). To distinguish between positive and negative formulas, we have to clone each construct $0, 1, \oplus, \otimes, \underset{w}{!}, \underset{c}{!}, \underset{e}{!}$ into itself: $0, 1, \oplus, \otimes, \underset{w}{!}, \underset{c}{!}, \underset{e}{!}$ and its dual: $\top, \bot, \&, ⅋, \underset{w}{?}, \underset{c}{?}, \underset{e}{?}$. The negation $\neg$ itself is cloned in two operations $\uparrow$ and $\downarrow$, each of them with a specific effect:

- $\uparrow$ transports the positive formulas into the negative formulas,
- $\downarrow$ transports the negative formulas into the positive formulas.

Note that the affine and exponential modalities do not change polarities themselves: this is a main difference with polarized logic. We use the letters P and Q for the positive formulas, the letters L and M for the negative formulas, and the letters Γ, Δ for the contexts of negative formulas. Formulas are constructed by the following grammar:

Positives	0	1	$\downarrow L$	$P \otimes Q$	$P \oplus Q$	$\underset{w}{!}P$	$\underset{c}{!}P$	$\underset{e}{!}P$
Negatives	$\bot$	$\top$	$\uparrow P$	$L ⅋ M$	$L \& M$	$\underset{w}{?}L$	$\underset{c}{?}L$	$\underset{e}{?}L$

Every positive formula P has a dual negative formula $P^\perp$, obtained by dualizing every logical construct appearing in the formula P. The sequent calculus in Figure 6 for the multiplicatives adapts Figure 1; Figure 7 for the additives adapts Figure 2. Figures 8, 9 and 10 for the resource modalities adapt Figures 3, 4 and 5.

Classical logic and polarized linear logic. Starting from Thielecke's work [43], Selinger designs the notion of *control category* in order to axiomatize the categorical semantics of classical logic [41]. Then, prompted by a nice completeness result discovered by Hofmann and Streicher [17], Selinger establishes a fundamental structure theorem, stating that every control category $\mathbf{P}$ is the continuation category $\mathcal{C}^\neg$ of a *response category* $\mathcal{C}$. Now, a response category $\mathcal{C}$ – where the monic requirement on the units (2) is relaxed – is the same thing as a model of multiplicative additive tensor logic, where the tensor $\otimes$ is *cartesian* and the tensor unit 1 is *terminal*.

Interestingly, a purely proof-theoretic analysis of classical logic leads exactly to the same conclusion. Exploiting Girard's work on polarities in LC [14], Laurent developed an extensive analysis of polarities in logic, incorporating classical logic, arena games and control categories [25, 26]. The main ingredient of his work is a logic called *polarized linear logic*, which happens to coincide with the multiplicative additive fragment of tensor logic, where the tensor product is *cartesian*, rather than monoidal. This fact appears clearly in the one-sided formulation of tensor logic. Note that the shift operators $\downarrow$ and $\uparrow$ of tensor logic are noted ! and ? in polarized linear logic, this leading to annoying confusions between negations and resource modalities. On the other hand, it should be observed that Laurent considered the multiplicative additive fragment of tensor logic in his PhD thesis [24] which he defined then as a "linear" version of polarized linear logic. Our point here is simply that one should proceed as in linear logic, and start from this linear version of polarized linear logic, rather than polarized linear logic itself. This starting point enables to get rid of the cartesian paradigm which haunts polarized logics since the early work by Girard on LC [14]. We sum up the difference between tensor logic and polarized linear logic in this very schematic table:

$$\frac{\vdash \Gamma, P \qquad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q}\ \text{Tensor} \qquad \frac{\vdash \Gamma_1, L, M, \Gamma_2, [P]}{\vdash \Gamma_1, L \parr M, \Gamma_2, [P]}\ \text{Par}$$

$$\frac{}{\vdash 1}\ \text{One} \qquad \frac{\vdash \Gamma, [P]}{\vdash \Gamma, \bot, [P]}\ \text{Bottom}$$

$$\frac{\vdash \Gamma, L}{\vdash \Gamma, \downarrow L}\ \text{Linear strengthening} \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, \uparrow P}\ \text{Linear dereliction}$$

$$\frac{}{\vdash P^{\perp}, P}\ \text{Axiom} \qquad \frac{\vdash \Gamma, P \qquad \vdash P^{\perp}, \Delta, [Q]}{\vdash \Gamma, \Delta, [Q]}\ \text{Cut}$$

Figure 6. Multiplicative tensor logic: one-sided presentation

$$\frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q}\ \text{Sum-Left} \qquad \frac{\vdash \Gamma, L, [P] \qquad \vdash \Gamma, M, [P]}{\vdash \Gamma, L \,\&\, M, [P]}\ \text{With}$$

$$\frac{\vdash \Gamma, Q}{\vdash \Gamma, P \oplus Q}\ \text{Sum-Right}$$

No introduction rule for the zero

$$\frac{}{\vdash \Gamma, \top, [P]}\ \text{Top}$$

Figure 7. Additive tensor logic: one-sided presentation

$$\frac{\vdash ?_e\Gamma, P}{\vdash ?_e\Gamma, !_e P}\ \text{Strengthening} \qquad \frac{\vdash \Gamma, L, [P]}{\vdash \Gamma, ?_e L, [P]}\ \text{Dereliction}$$

$$\frac{\vdash \Gamma, [P]}{\vdash \Gamma, ?_e L, [P]}\ \text{Weakening} \qquad \frac{\vdash \Gamma, ?_e L, ?_e L, [P]}{\vdash \Gamma, ?_e L, [P]}\ \text{Contraction}$$

Figure 8. Exponential modality: one-sided presentation

$$\frac{\vdash ?_w\Gamma, P}{\vdash ?_w\Gamma, !_w P}\ \text{Strengthening} \qquad \frac{\vdash \Gamma, L, [P]}{\vdash \Gamma, ?_w L, [P]}\ \text{Dereliction}$$

$$\frac{\vdash \Gamma, [P]}{\vdash \Gamma, ?_w L, [P]}\ \text{Weakening}$$

Figure 9. Affine modality: one-sided presentation

$$\frac{\vdash ?_c\Gamma, P}{\vdash ?_c\Gamma, !_c P}\ \text{Strengthening} \qquad \frac{\vdash \Gamma, L, [P]}{\vdash \Gamma, ?_c L, [P]}\ \text{Dereliction}$$

$$\frac{\vdash \Gamma, ?_c L, ?_c L, [P]}{\vdash \Gamma, ?_c L, [P]}\ \text{Contraction}$$

Figure 10. Relevant modality: one-sided presentation

Tensor logic	$\otimes$	is monoidal
	$\neg$	is tensorial
Polarized linear logic	$\otimes$	is cartesian
	$\neg$	is tensorial

One advantage of the approach is that every resource modality (6) on a dialogue category $\mathcal{C}$ induces a structure of dialogue category on the category $\mathcal{M}$, where negation is defined as

$$F^{op} \circ \neg \circ U \quad : \quad \mathcal{M} \to \mathcal{M}^{op}.$$

Note that the self-adjunction induced by the tensorial negation on the category $\mathcal{M}$ may be alternatively described as the composite of the three adjunctions

$$\begin{tikzcd} \mathcal{M} \arrow[r, bend left, "U"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "F"] \arrow[r, bend left, "\neg"] \arrow[r, phantom, "\perp"] & \mathcal{C}^{op} \arrow[l, bend left, "\neg"] \arrow[r, bend left, "F^{op}"] \arrow[r, phantom, "\perp"] & \mathcal{M}^{op} \arrow[l, bend left, "U^{op}"] \end{tikzcd}$$

In particular, when the resource modality (6) is exponential, the monoidal structure of the category $\mathcal{M}$ is provided by the cartesian product, and one thus obtains a model of polarized linear logic. This construction should be thought as a polarized version of the familiar construction in linear logic of a cartesian closed category from an $*$-autonomous category equipped with a resource modality.

Linear logic. The continuation monad $A \mapsto \overset{O\,P}{\neg\neg} A$ of game semantics lifts an Opponent-starting game A with an Opponent move $\neg_O$ followed by a Player move $\neg_P$. Now, it appears that the Blass problem mentioned in §1 arises precisely because the monad is strong, but not commutative, see [35, 31] for details. Motivated by this key observation, the first autho developed asynchronous game semantics in order to establish that innocent strategies are *positional* [30]. This positionality result enables then to *identify* the two canonical strategies (5) and to obtain in this way a game-theoretic model of full propositional linear logic. This approach leads eventually to a fully complete model of linear logic, based on an appropriate winning condition on strategies, described in [31, 32].

It appears that this game-theoretic construction has a nice categorical counterpart. It is well-known since the work by Power and Robinson [40] that the Kleisli category $\mathcal{C}_T$ associated to a monad T inherits a *premonoidal* structure from the monoidal structure of the category $\mathcal{C}$, when the monad T is strong. Moreover, when the monad T is not only strong, but also commutative, the premonoidal structure on the Kleisli category $\mathcal{C}_T$ becomes monoidal. A conceptual explanation for this phenomenon is that a commutative monad is the same thing as a monoidal monad, in a lax sense [20, 34]. So, when the continuation monad T is not only strong, but also commutative, its Kleisli category $\mathcal{C}_T$ is symmetric monoidal. Now, Hasegawa observed a much stronger property [16]:

Proposition 1 *Given a dialogue category $\mathcal{C}$, the following are equivalent:*

- *the continuation monad is commutative,*
- *the continuation monad is idempotent, this meaning that the multiplication of the monad*

$$\mu_A \quad : \quad \neg\neg\neg\neg\, A \quad \longrightarrow \quad \neg\neg\, A$$

  is an isomorphism, for every object A of the category,

- *the morphisms*
$$\eta_{\neg A} \quad : \quad \neg\, A \quad \leftrightarrows \quad \neg\neg\neg\, A \quad : \quad \neg\, \eta_A$$
are inverse morphisms, for every object A of the category $\mathcal{C}$,

- *the Kleisli category $\mathcal{C}_T$ equipped with the premonoidal structure inherited from monoidal structure of the dialogue category is $*$-autonomous.*

This construction provides in fact a categorification of Girard's phase space semantics [13]. It should be observed in particular that the Kleisli category $\mathcal{C}_T$ is equivalent in that case to the full subcategory of $\mathcal{C}$ consisting of the negated objects (that is, of the form $\neg A$). This result demonstrates that linear logic is essentially the same thing as tensor logic where the tensorial negation is commutative (in the sense that it induces a commutative continutation monad).

Linear logic	$\otimes$	is monoidal
	$\neg$	is commutative

We now develop this idea and show that any model of full propositional tensor logic, where the continuation monad is commutative, induces a model of linear logic on the Kleisli category $\mathcal{C}_T$ of the continuation monad. The idea is to start from the adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F_T"] \arrow[r, phantom, "\perp"] & \mathcal{C}_T \arrow[l, bend left, "G_T"] \end{tikzcd} \tag{9}$$

between the category $\mathcal{C}$ and its Kleisli category $\mathcal{C}_T$. Since we are considering the case of a commutative continuation monad T, the adjunction is symmetric monoidal. This establishes already that when the continuation monad is commutative,

Lemma 1 *Every exponential modality on the category $\mathcal{C}$ induces an exponential modality on the $*$-autonomous category $\mathcal{C}_T$, and thus a model of multiplicative exponential linear logic.*

Proof: The proof is simply based on the fact that the two symmetric monoidal adjunctions

$$\begin{tikzcd} \mathcal{M} \arrow[r, bend left, "U"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "F"] \arrow[r, bend left, "F_T"] \arrow[r, phantom, "\perp"] & \mathcal{C}_T \arrow[l, bend left, "G_T"] \end{tikzcd}$$

compose as a symmetric monoidal adjunction. Now, since the monoidal structure of the category $\mathcal{M}$ is provided by its finite product, this adjunction defines an exponential modality on the $*$-autonomous category $\mathcal{C}_T$. $\square$

Next, we show how to interpret the additive structure of linear logic, from a model of tensor logic with sums. The recipe is based on a folklore result in category theory, which states that the Kleisli category of a category with finite coproducts is also a category with finite coproducts. More precisely,

Proposition 2 *Suppose that T is a monad on a category $\mathcal{C}$ with finite coproducts. In that case, the Kleisli category $\mathcal{C}_T$ is also equipped with finite coproducts.*

From this, it follows that the Kleisli category $\mathcal{C}_T$ has finite coproducts, when the dialogue category $\mathcal{C}$ has finite coproducts. Moreover, the Kleisli category $\mathcal{C}_T$ has also finite products defined by duality as

$$A \,\&\, B \stackrel{\text{def}}{=} \neg(\neg A \oplus \neg B)$$

when the continuation monad is commutative. From this, it follows that

Theorem 1 *Suppose that $\mathcal{C}$ is a model of propositional tensor logic (multiplicative, additive, exponential) where the continuation monad $\neg\neg$ is commutative. Then, the Kleisli category $\mathcal{C}_T$ defines a model of propositional linear logic (multiplicative, additive, exponential).*

Remark. Suppose given a dialogue category $\mathcal{C}$ equipped with an exponential modality described as a comonad $!_e$. Suppose moreover that the continuation monad induced by the dialogue category is commutative. We have seen in Proposition 1 that the continuation monad is also idempotent. From this follows that the Kleisli category $\mathcal{C}_T$ of the monad is equivalent to its category $\mathcal{C}^T$ of Eilenberg-Moore algebras. This leads to two alternative descriptions of the adjunction (9). In the first formulation, the objects of the Kleisli category $\mathcal{C}_T$ are defined as the objects of the category $\mathcal{C}$, and F_T is defined as the identity on objects whereas G_T is defined as double negation on objects. One obtains the formula

$$!\,A \quad = \quad !_e\,\neg\neg\,A$$

for the exponential modality of linear logic on the $*$-autonomous category $\mathcal{C}_T$. In the second formulation, the objects of the Kleisli category $\mathcal{C}_T$ are defined as the algebras of the category $\mathcal{C}$, and F_T is defined as double negation whereas G_T is defined as the identity on objects. One obtains the formula

$$!\,A \quad = \quad \neg\neg\,!_e\,A$$

for the exponential modality of linear logic on the $*$-autonomous category $\mathcal{C}^T$. Despite their superficial difference, it should be stressed that the two constructions of the exponential modality ! are equivalent when the continuation monad is commutative.

Three translations of linear logic into tensor logic. We describe three syntactical translations of linear logic into tensor logic, whose difference lies in the number of negations introduced between the logical connectives of linear logic. In particular, the three translations are isomorphic when the continuation monad is commutative, or equivalently, when the continuation monad is idempotent. So, they provide in that case three alternative but isomorphic descriptions of the categorical translation implemented by the Kleisli construction in Theorem 1.

First translation: negative translation. The first translation provides a direct and somewhat naive syntactical counterpart of the Kleisli construction. The translation is a variant of the Gödel-Gentzen negative translation of classical logic (LK) into intuitionistic logic (LJ). The idea is to translate every formula A of linear logic as its negation $(A)^N$

$$
\begin{array}{rcl@{\qquad}rcl}
(\top)^N & \stackrel{\mathrm{def}}{=} & 0 & (0)^N & \stackrel{\mathrm{def}}{=} & \neg\, 0 \\
(\bot)^N & \stackrel{\mathrm{def}}{=} & 1 & (1)^N & \stackrel{\mathrm{def}}{=} & \neg\, 1 \\
(A \,\&\, B)^N & \stackrel{\mathrm{def}}{=} & (A)^N \oplus (B)^N & (A \oplus B)^N & \stackrel{\mathrm{def}}{=} & \neg\,(\neg(A)^N \oplus \neg(B)^N) \\
(A \parr B)^N & \stackrel{\mathrm{def}}{=} & (A)^N \otimes (B)^N & (A \otimes B)^N & \stackrel{\mathrm{def}}{=} & \neg\,(\neg(A)^N \otimes \neg(B)^N) \\
(?A)^N & \stackrel{\mathrm{def}}{=} & \oc_e (A)^N & (!A)^N & \stackrel{\mathrm{def}}{=} & \neg\, \oc_e \neg(A)^N
\end{array}
$$

Figure 11. Negative translation of linear logic into tensor logic (two-sided).

in tensor logic. This negative translation on formulas is described in Figure 11. Every sequent

$$\vdash \quad A_1, \ldots, A_k$$

of linear logic is then translated as the sequent

$$(A_1)^N, \ldots, (A_k)^N \quad \vdash$$

of tensor logic. One establishes that

Proposition 3 *Every proof of the sequent* $\vdash A_1, \ldots, A_k$ *in linear logic induces a proof of the sequent* $(A_1)^N, \ldots, (A_k)^N \vdash$ *in tensor logic.*

The proof of Proposition 3 is performed by structural induction on the derivation tree of the sequent $\vdash A_1, \ldots, A_k$ in linear logic. Every step of the derivation tree in linear logic is translated:

- in one step in the case of the connectives & and $\parr$, and the constants $\top$ and $\bot$,
- in two steps in the case of the constants 0 and 1,
- in three steps in the case of the connectives $\oplus$ and $\otimes$.

Typically, the introduction of the connective $\parr$ in linear logic

$$\frac{\vdash\ \Gamma\ ,\ A\ ,\ B}{\vdash\ \Gamma\ ,\ A \parr B}$$

is translated in one step as

$$\frac{(\Gamma)^N\ ,\ (A)^N\ ,\ (B)^N \quad \vdash}{(\Gamma)^N\ ,\ (A)^N \otimes (B)^N \quad \vdash}$$

in tensor logic. On the other hand, the introduction of the connective $\otimes$ in linear logic

$$\frac{\vdash\ \Gamma\ ,\ A \qquad \vdash\ \Delta\ ,\ B}{\vdash\ \Gamma\ ,\ \Delta\ ,\ A \otimes B}$$

is translated in three steps as

$$\frac{\dfrac{\dfrac{(\Gamma)^N\ ,\ (A)^N\ \vdash}{(\Gamma)^N\ \vdash\ \neg(A)^N} \qquad \dfrac{(\Delta)^N\ ,\ (B)^N\ \vdash}{(\Delta)^N\ \vdash\ \neg(B)^N}}{(\Gamma)^N\ \vdash\ \neg(A)^N \otimes \neg(B)^N}}{(\Gamma)^N\ ,\ \neg(\neg(A)^N \otimes \neg(B)^N)\ \vdash}$$

The negative translation may be equivalently described as a translation of linear logic into tensor logic, seen from the one-sided point of view, rather than from the two-sided point of view. Formulated in this way, the negative translation translates a formula A of linear logic into a formula $(A)^N$ of tensor logic by introducing a sufficient number of shift operators $\downarrow$ or $\uparrow$ between the connectives of the formula A, as explained in Figure 12. Of course, every sequent

$$\vdash \quad A_1, \ldots, A_k$$

of linear logic is then translated as the sequent

$$\vdash \quad (A_1)^N, \ldots, (A_k)^N$$

of tensor logic, formulated in the one-sided point of view. A converse of Proposition 3 may be established, where we keep the one-sided formulation for clarity's sake:

Proposition 4 *Every proof of* $\vdash (A_1)^N, \ldots, (A_k)^N$ *in tensor logic induces a proof of the sequent* $\vdash A_1, \ldots, A_k$ *in linear logic.*

The proof is performed by structural induction on the derivation tree of $\vdash (A_1)^N, \ldots, (A_k)^N$ in tensor logic. Every step of the derivation tree of tensor logic is interpreted in

- no step for the shift connectives $\downarrow$ or $\uparrow$,
- one step for all the other connectives

in linear logic. So, the translation from tensor logic is to remove all the logical steps introducing a shift operator $\downarrow$ or $\uparrow$, this leading to a proof of the sequent $\vdash A_1, \ldots, A_k$ of linear logic. This last point follows from the previous observation that the negative translation, seen from the one-sided point of view, consists only in introducing a sufficient number of shift operators $\downarrow$ or $\uparrow$ between the connectives of the formula A of linear logic.

Remark. The pair of back and forth translations formulated in Propositions 3 and 4 establishes that tensor logic is a refinement of linear logic, in the sense that every proof in tensor logic translates (in a canonical way) as a proof of linear logic, obtained by removing the shift operators $\downarrow$ or $\uparrow$ from the formulas and from the proofs, and conversely, every proof of linear logic lifts (in a non canonical way) to a proof of tensor logic obtained by adding a sufficient number of shift operators $\downarrow$ or $\uparrow$ in the formulas and in the proofs. Let us illustrate here the fact that the lifting from linear logic to tensor logic is far from canonical. It is customary to consider that the two proofs of linear logic

$$\dfrac{\dfrac{\vdash A \qquad \vdash B\ ,\ C}{\vdash A \otimes B\ ,\ C} \qquad \vdash D}{\vdash A \otimes B\ ,\ C \otimes D} \qquad\qquad \dfrac{\vdash A \qquad \dfrac{\vdash B\ ,\ C \qquad \vdash D}{\vdash B\ , C \otimes D}}{\vdash A \otimes B\ ,\ C \otimes D} \tag{10}$$

are equal, because they are interpreted as the same morphism in any $*$-autonomous category. The negative translation translates the proof on the left-hand side as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash (A)^N}{\vdash \downarrow (A)^N} \qquad \dfrac{\vdash (B)^N\ ,\ (C)^N}{\vdash \downarrow (B)^N\ ,\ (C)^N}}{\vdash \downarrow (A)^N \otimes \downarrow (B)^N\ ,\ (C)^N}}{\vdash \uparrow (\downarrow (A)^N \otimes \downarrow (B)^N)\ ,\ (C)^N}}{\vdash \uparrow (\downarrow (A)^N \otimes \downarrow (B)^N)\ ,\ \downarrow (C)^N} \qquad \dfrac{\vdash (D)^N}{\vdash \downarrow (D)^N}}{\vdash \uparrow (\downarrow (A)^N \otimes \downarrow (B)^N)\ ,\ \downarrow (C)^N \otimes \downarrow (D)^N}}{\vdash \uparrow (\downarrow (A)^N \otimes \downarrow (B)^N)\ ,\ \uparrow (\downarrow (C)^N \otimes \downarrow (D)^N)}$$

$$
\begin{array}{rcl@{\qquad}rcl}
(\top)^N & \stackrel{\text{def}}{=} & \top & (0)^N & \stackrel{\text{def}}{=} & \uparrow 0 \\
(\bot)^N & \stackrel{\text{def}}{=} & \bot & (1)^N & \stackrel{\text{def}}{=} & \uparrow 1 \\
(A \,\&\, B)^N & \stackrel{\text{def}}{=} & (A)^N \,\&\, (B)^N & (A \oplus B)^N & \stackrel{\text{def}}{=} & \uparrow (\downarrow (A)^N \oplus \downarrow (B)^N\,) \\
(A \parr B)^N & \stackrel{\text{def}}{=} & (A)^N \parr (B)^N & (A \otimes B)^N & \stackrel{\text{def}}{=} & \uparrow (\downarrow (A)^N \otimes \downarrow (B)^N\,) \\
(?A)^N & \stackrel{\text{def}}{=} & ¿\,(A)^N & (!A)^N & \stackrel{\text{def}}{=} & \uparrow ¡ \downarrow (A)^N
\end{array}
$$

Figure 12. Negative translation of linear logic into tensor logic (one-sided)

whereas it translates the proof on the right-hand side as

$$
\dfrac{\dfrac{\dfrac{}{\vdash\ (A)^N}}{\vdash\ \downarrow (A)^N} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdash\ (B)^N\ ,\ (C)^N}{\vdash\ (B)^N\ ,\downarrow (C)^N} \qquad \dfrac{\vdash\ (D)^N}{\vdash\ \downarrow (D)^N}}{\vdash\ (B)^N\ ,\downarrow (C)^N \otimes \downarrow (D)^N}}{\vdash\ (B)^N\ ,\uparrow (\downarrow (C)^N \otimes \downarrow (D)^N)}}{\vdash\ \downarrow (B)^N\ ,\uparrow (\downarrow (C)^N \otimes \downarrow (D)^N)}}{\dfrac{\vdash\ \downarrow (A)^N \otimes \downarrow (B)^N\ ,\ \uparrow (\downarrow (C)^N \otimes \downarrow (D)^N)}{\vdash\ \uparrow (\downarrow (A)^N \otimes \downarrow (B)^N)\ ,\ \uparrow (\downarrow (C)^N \otimes \downarrow (D)^N)}}
$$

These two proofs of tensor logic should not be considered equal in tensor logic, because they are interpreted as different morphisms in a typical dialogue category like the category **Games** defined in §4. This illustrates the fact that the same proof of linear logic may be lifted in several ways to tensor logic. Each translation implements a particular scheduling for the exploration of the logical connectives, nicely reflected as a sequential strategy in game semantics.

Second translation: linear translation. The second translation is called the linear translation because we believe that it reflects the familiar structure of linear logic: in particular, the translation introduces one shift operator (at least) between any two connectives of linear logic, which is not the case with the negative translation. The linear translation is mainly motivated by an unpleasant aspect of the negative translation: the fact that it is not symmetric, in the sense that the negative translation $(A^*)^N$ of the dual A^* of a formula A of linear logic... is not the dual of the formula $(A)^N$ in tensor logic. In order to achieve such a symmetric translation, we need to separate the formulas of linear logic in two classes: the positive formulas of the form

$$A \oplus B \quad | \quad 0 \quad | \quad A \otimes B \quad | \quad 1 \quad | \quad !A$$

and the negative formulas of the form

$$A \,\&\, B \quad | \quad \top \quad | \quad A \parr B \quad | \quad \bot \quad | \quad ?A.$$

The linear translation is then performed by translating the units 0 and 1 as the formulas $(0)^L = \downarrow\uparrow 0$ and $(1)^L = \downarrow\uparrow 1$, and dually, the units $\top$ and $\bot$ as the formulas $(\top)^L = \uparrow\downarrow\top$ and $(\top)^L = \uparrow\downarrow\bot$. And then, by carefully applying the table below for each connective of linear logic. Note in particular that a positive formula of linear logic is translated as a positive formula of tensor logic, whereas a negative formula is translated as a negative formula of tensor logic.

A	B	$(A \otimes B)^L$	$(A \parr B)^L$	$(A \oplus B)^L$	$(A \,\&\, B)^L$	$(!A)^L$	$(?A)^L$
$+$	$+$	$\downarrow\uparrow A^L \otimes \downarrow\uparrow B^L$	$\uparrow A^L \parr \uparrow B^L$	$\downarrow\uparrow A^L \oplus \downarrow\uparrow B^L$	$\uparrow A^L \,\&\, \uparrow B^L$	$!_e\downarrow\uparrow A^L$	$?_e\uparrow A^L$
$+$	$-$	$\downarrow\uparrow A^L \otimes \downarrow B^L$	$\uparrow A^L \parr \uparrow\downarrow B^L$	$\downarrow\uparrow A^L \oplus \downarrow B^L$	$\uparrow A^L \,\&\, \uparrow\downarrow B^L$		
$-$	$+$	$\downarrow A^L \otimes \downarrow\uparrow B^L$	$\uparrow\downarrow A^L \parr \uparrow B^L$	$\downarrow A^L \oplus \downarrow\uparrow B^L$	$\uparrow\downarrow A^L \,\&\, \uparrow B^L$	$!_e\downarrow A^L$	$?_e\uparrow\downarrow A^L$
$-$	$-$	$\downarrow A^L \otimes \downarrow B^L$	$\uparrow\downarrow A^L \parr \uparrow\downarrow B^L$	$\downarrow A^L \oplus \downarrow B^L$	$\uparrow\downarrow A^L \,\&\, \uparrow\downarrow B^L$		

Third translation: focalized translation. This last translation is a variant of the linear translation: in particular, both translations are symmetric. The translation is called focalized because it introduces as few shift operators as possible between the connectives of linear logic. Each connective of linear logic becomes part of a cluster of positive or negative connectives after translation in tensor logic, this reflecting the focalization property of linear logic, noticed for the first time by Andreoli [5]. Note that the focusing translation starts by translating the positive units 0 and 1 as themselves, and similarly for the negative units $\top$ and $\bot$.

A	B	$(A \otimes B)^F$	$(A \parr B)^F$	$(A \oplus B)^F$	$(A \,\&\, B)^F$	$(!A)^F$	$(?A)^F$
$+$	$+$	$A^F \otimes B^F$	$\uparrow A^F \parr \uparrow B^F$	$A^F \oplus B^F$	$\uparrow A^F \,\&\, \uparrow B^F$	$!_e A^F$	$?_e\uparrow A^F$
$+$	$-$	$A^F \otimes \downarrow B^F$	$\uparrow A^F \parr B^F$	$A^F \oplus \downarrow B^F$	$\uparrow A^F \,\&\, B^F$		
$-$	$+$	$\downarrow A^F \otimes B^F$	$A^F \parr \uparrow B^F$	$\downarrow A^F \oplus B^F$	$A^F \,\&\, \uparrow B^F$	$!_e\downarrow A^F$	$?_e A^F$
$-$	$-$	$\downarrow A^F \otimes \downarrow B^F$	$A^F \parr B^F$	$\downarrow A^F \oplus \downarrow B^F$	$A^F \,\&\, B^F$		

Free finite coproducts. In several important situations arising in game semantics, one finds a dialogue category $\mathcal{C}$ with finite products, but without finite coproducts. In that case, it is tempting to add these coproducts in a free way, by using the *family construction* described by Abramsky and McCusker in [4]. Recall that the category $Fam(\mathcal{C})$ is defined as follows:

- its objects are the families $\{A_i \mid i \in I\}$ of objects of $\mathcal{C}$, where I is a finite set,
- its morphisms from $\{A_i \mid i \in I\}$ to $\{B_j \mid j \in J\}$ are the pairs consisting of a reindexing function $f : I \to J$ together with a family of morphisms $\{f_i : A_i \to B_{f(i)} \mid i \in I\}$ of the category $\mathcal{C}$.

It is folklore that this family construction defines the free completion $Fam(\mathcal{C})$ under finite coproducts generated by the category $\mathcal{C}$. This family construction defines a 2-monad on the 2-category of categories, functors and natural transformations. In fact, Hyland and Power use this 2-monad as a concrete illustration of their notion of symmetric pseudo-commutative 2-monad [19]. In particular, they deduce from this property of the 2-monad Fam that it distributes with the 2-monad constructing the free symmetric monoidal category generated by a category. From this follows that

1. the category $Fam(\mathcal{C})$ inherits the symmetric monoidal structure from the category $\mathcal{C}$,
2. finite coproducts in $Fam(\mathcal{C})$ distributes with the tensor product,
3. the 2-functor Fam preserves symmetric monoidal adjunction.

Typically, the tensor product of $A = \{A_i | i \in I\}$ and $B = \{B_j | j \in J\}$ is defined as

$$A \otimes B \quad \stackrel{\text{def}}{=} \quad \{A_i \otimes B_j \mid (i,j) \in I \times J\}.$$

Moreover, Abramsky and McCusker show that the family construction transports categories with finite product into categories with finite products [4]. From all this follows that the family construction preserves affine and exponential modalities, and that the finite products of the category $\mathcal{C}$ (when they exist) lift to the category $Fam(\mathcal{C})$. Typically, the cartesian product of $A = \{A_i | i \in I\}$ and $B = \{B_j | j \in J\}$ is defined as

$$A \,\&\, B \quad \stackrel{\text{def}}{=} \quad \{A_i \,\&\, B_j \mid (i,j) \in I \times J\}.$$

Now, suppose that the dialogue category $\mathcal{C}$ has finite products, noted

$$\mathop{\&}_{i \in I} A_i$$

for a family $(A_i)_{i \in I}$ of objects of the category $\mathcal{C}$, indexed by the finite set I. In that case, it is equip the category $Fam(\mathcal{C})$ with a tensorial negation defined as

$$\neg A \quad = \quad \{ \mathop{\&}_{i \in I} (\neg A_i) \}. \tag{11}$$

for every object $A = \{A_i \,|\, i \in I\}$ of the category $Fam(\mathcal{C})$. This establishes that

Proposition 5 *Suppose that the category $\mathcal{C}$ is a dialogue category with finite products. Then, the category $Fam(\mathcal{C})$ is a dialogue category with finite products.*

Putting all this together, one obtains the following property:

Theorem 2 *Suppose that the category $\mathcal{C}$ is a dialogue category with finite products, equipped with an affine, a relevant and an exponential resource modality. Then, the category $Fam(\mathcal{C})$ is a model of propositional tensor logic (multiplicative, additive, affine, relevant, exponential).*

4 Payoff Conway games

In this section and in the next one, we construct a simple game semantics of tensor logic, starting from the graph-theoretic notion of Conway game introduced by Joyal in his pioneering work on categories of games [21]. The main idea of our construction is to refine the original definition of Conway games with a notion of *payoff* on positions, in order to reflect the resource modalities of tensor logic. The purpose of this section is to construct a dialogue category **Games** of Conway games with payoff (see Proposition 10) while the purpose of the next section is to interpret the resource modalities in this category.

Conway games. A *Conway game* is defined as a rooted graph (V, E, λ) consisting of

- a set V of vertices called the *positions* of the game,
- a set $E \subset V \times V$ of edges called the *moves* of the game,
- a function $\lambda : E \to \{-1, +1\}$ indicating whether a move belongs to Opponent (-1) or Proponent $(+1)$,

The root of the game A will be denoted $\star_A$. A Conway game is called *negative* (resp. *positive*) when all the moves starting from the root belong to Opponent (resp. Proponent).

Path and play. A *play* $m_1 \cdot m_2 \cdot \ldots \cdot m_{k-1} \cdot m_k$ of a Conway game A is a path starting from the root $\star_A$:

$$\star_A \xrightarrow{m_1} x_1 \xrightarrow{m_2} \ldots \xrightarrow{m_{k-1}} x_{k-1} \xrightarrow{m_k} x_k \tag{12}$$

A play (12) is *alternating* when:

$$\forall i \in \{1, \ldots, k-1\}, \qquad \lambda_A(m_{i+1}) = -\lambda_A(m_i).$$

We note Play_A the set of plays of a game A.

Strategies. A *strategy* σ of a Conway game A is defined as a non empty set of *alternating plays* of even length such that

- every non empty play starts with an Opponent move,
- σ is closed by even length prefix: for all plays s and for all moves m, n,

$$s \cdot m \cdot n \in \sigma \quad \text{implies} \quad s \in \sigma;$$

- σ is deterministic: for all plays s, and for all moves m, n, n',

$$s \cdot m \cdot n \in \sigma \quad \text{and} \quad s \cdot m \cdot n' \in \sigma \quad \text{implies} \quad n = n'.$$

Note that our notion of strategy is *partial* because a strategy does not necessarily have to answer to an Opponent move. We write $\sigma : A$ to indicate that σ is strategy over the game A.

Remark. It is worth observing that the definition of Conway game does not require that the plays of the game are alternating. The notion of alternation between Opponent and Proponent only appears at the level of strategies (i.e. proofs) and not at the level of games (i.e. formulas). This corresponds to the intuition that a game describes a fairly liberal space of interaction whereas a strategy implements regulated executions.

Dual. Every Conway game A induces a *dual* game A^* obtained simply by reversing the polarity of moves. Formally speaking, $A^* = (V_{A^*}, E_{A^*}, \lambda_{A^*})$ is defined by

- $V_{A^*} = V_A$;
- $E_{A^*} = E_A$;
- $\lambda_{A^*} = -\lambda_A$.

Tensor product. The tensor product $A \otimes B$ of two Conway games A and B is essentially the asynchronous product of the two underlying graphs. More formally, it is defined as:

- its positions are the pairs (x, y) noted $x \otimes y$ with $\star_{A\otimes B} = \star_A \otimes \star_B$, that is

$$V_{A\otimes B} = V_A \times V_B,$$

- its moves are of two kinds :

$$x \otimes y \rightarrow \begin{cases} z \otimes y \text{ if } x \rightarrow z \text{ in the game } A \\ x \otimes z \text{ if } y \rightarrow z \text{ in the game } B, \end{cases}$$

- the polarity of a move in the game $A \otimes B$ is inherited from the polarity of the underlying move in the game A or B.

The Conway game with a unique position $\star$ and no move will be denoted 1. It is the neutral element of the tensor product. Observe that every play s of the game $A \otimes B$ may be seen as the interleaving of a play $s_{|A}$ of the game A and a play $s_{|B}$ of the game B.

Composition. The composite of two strategies is defined by "parallel composition plus hiding", a formal description of composition meaningful in game semantics and more generally in any compact closed category. We proceed as in [29, 15] and define an interaction u on the three games A, B, C as a play of the game $A \otimes B \otimes C$, what we write $u \in int_{ABC}$. Note that a word u on the alphabet $E_A + E_B + E_C$ is an element of int_{ABC} precisely when the projection of u on each component $E_A + E_B$ and $E_B + E_C$ and $E_A + E_C$ defines a play in the game $A^* \otimes B$, $B^* \otimes C$ and $A^* \otimes C$, respectively. The composite $\sigma; \tau$ of two strategies $\sigma : A^* \otimes B$ and $\tau : B^* \otimes C$ is then defined as

$$\sigma; \tau = \{u_{|A^* \otimes C} \mid u \in int_{ABC}, u_{|A^* \otimes B} \in \sigma, u_{|B^* \otimes C} \in \tau\}$$

One then checks that the composite $\sigma; \tau$ defines a strategy of the game $A^* \otimes B$.

Identity morphism. The identity morphism id_A on a game A is defined as the copycat strategy on the game $A^* \otimes A$ described by Joyal [21]. The idea is that for every Opponent move in one of the component A^* or A, the copycat strategy responds with the dual move in the other component. Formally speaking, the identity is defined as

$$id_A \quad \stackrel{\text{def}}{=} \quad \{s \in \mathrm{Play}^{even}_{A_1^* \otimes A_2} \mid \forall t \preceq^{even} s \ , \ t_{|A_1} = t_{|A_2}\}$$

where one uses the tags 1 and 2 in order to distinguish between the two occurrences of the game A and where the exponent *even* restricts the prefix relation to the paths of even length.

The category of Conway games. The category **Conway** has Conway games as objects, and strategies σ of $A^* \otimes B$ as morphisms $\sigma : A \rightarrow B$. The resulting category **Conway** is compact-closed in the sense of [22] with units $\eta_A : 1 \rightarrow A \otimes A^*$ and counits $\varepsilon_A : A^* \otimes A \rightarrow 1$ defined as copycat strategies. Interestingly, all we need here is that **Conway** is symmetric monoidal closed, with linear implication defined as

$$A \multimap B \quad \stackrel{\text{def}}{=} \quad A^* \otimes B.$$

In particular, the full subcategory **Conway**$^-$ of negative Conway games is no longer compact closed, but still, it inherits a linear implication of **Conway**. The reason is that the embedding functor from **Conway**$^-$ to **Conway** is full and faithful (by definition) and has a right adjoint: the functor which transports every Conway game A to the negative game A^- obtained by removing all Proponent moves starting from the root. This functor is full (but not faithful) from **Conway** to **Conway**$^-$, and makes **Conway**$^-$ a coreflective subcategory of **Conway**. This is enough to deduce the linear implication of the category **Conway**$^-$ from the linear implication of the category **Conway**, as follows:

$$A \multimap B \quad \stackrel{\text{def}}{=} \quad (A \mathrel{-\!\!\bullet} B)^-$$

This general fact is established in the following proposition.

Proposition 6 *Suppose that $(\mathcal{C}, \otimes, \mathrel{-\!\!\bullet})$ is a symmetric monoidal closed category and that $(\mathcal{D}, \otimes)$ is a symmetric monoidal category. Suppose that there exists a monoidal adjunction $U \dashv F : \mathcal{D} \to \mathcal{C}$ where the functor U is full and faithful. In that case, the category $\mathcal{D}$ is symmetric monoidal closed with linear implication defined as*

$$A \multimap B \quad \stackrel{\text{def}}{=} \quad F(U(A) \mathrel{-\!\!\bullet} U(B))$$

for all objects A, B of the category $\mathcal{D}$.

Proof: Recall that a monoidal adjunction $U \dashv F$ is the same thing as an adjunction where the left adjoint functor U is strong monoidal. The fact that $A \multimap B$ defines a linear implication is deduced from the following series of natural bijections:

$$\begin{array}{rcll}
\mathcal{D}(B, A \multimap C) & \cong & \mathcal{D}(B, F(U(A) \mathrel{-\!\!\bullet} U(C))) & \\
 & \cong & \mathcal{C}(U(B), U(A) \mathrel{-\!\!\bullet} U(C)) & \text{adjunction } U \dashv F \\
 & \cong & \mathcal{C}(U(A) \otimes U(B), U(C)) & \text{linear implication in } \mathcal{C} \\
 & \cong & \mathcal{C}(U(A \otimes B), U(C)) & U \text{ is strong monoidal} \\
 & \cong & \mathcal{D}(A \otimes B, C) & U \text{ is full and faithful}
\end{array}$$

□

This established that

Proposition 7 *The category* **Conway**$^-$ *is symmetric monoidal closed.*

Our next step is to refine our definition of Conway game with a notion of payoff function on positions. As we will see, this leads to the definition of a self-dual category **Payoff** of payoff Conway games and winning strategies. This category extends the category **Conway** in the sense that the category **Conway** may be identified as the full subcategory of games with only neutral positions in the category **Payoff**.

Payoff Conway games. A *payoff Conway game* is a Conway game $A = (V_A, E_A, \lambda_A)$ equipped with a payoff function (defined on positions)

$$\kappa_A : V_A \to \{-1, 0, +1\}.$$

A position is called *winning* when $\kappa_A(x) \in \{0, +1\}$. Intuitively, the value -1 denotes a winning position for Opponent, the value $+1$ denotes a winning position for Proponent, and the value 0 denotes a "neutral" position.

$\otimes$	-1	0	$+1$
-1	-1	-1	-1
0	-1	0	$+1$
$+1$	-1	$+1$	$+1$

$\multimap$	-1	0	$+1$
-1	$+1$	$+1$	$+1$
0	-1	0	$+1$
$+1$	-1	-1	$+1$

Figure 13. Payoff tables of the tensor product and the linear implication.

Tensor product and linear implication of payoff games. We now extend the tensor product and the linear implication to payoff Conway games. As the payoff is positional, it is sufficient to provide a “truth table” (cf. Figure 13) for each connective $\otimes$ and $\multimap$. The construction of the two payoff tables is guided by the intuition that $\otimes$ corresponds to a boolean conjunction on payoffs, that $\multimap$ corresponds to a boolean implication, that -1 corresponds to false, that $+1$ corresponds to true, and that 0 corresponds to a third (and neutral) truth value. Note that these tables are simplified versions of the payoff tables appearing in [31, 32].

The payoff Conway game $A \otimes B$ is thus defined as the underlying Conway game $A \otimes B$, equipped with the payoff function

$$\kappa_{A\otimes B}(x \otimes y) \quad = \quad \kappa_A(x) \otimes \kappa_B(y)$$

and the payoff Conway game $A \multimap B$ is defined as the underlying Conway game $A \multimap B$, equipped with the payoff function

$$\kappa_{A\multimap B}(x \multimap y) \quad = \quad \kappa_A(x) \multimap \kappa_B(y).$$

We assign payoff 0 to the unique position $\star$ of the game 1.

Winning strategies. With the traditional notion of strategy between Conway games, every negative game has a unique morphism to the unit game 1. In the model of tensor logic constructed below, we use the payoff function in order to define a notion of *winning strategy*, which enables us to distinguish between *affine games* (whose payoff at the root is 0) and *linear games* (whose payoff at the root is $+1$).

A strategy σ on a payoff Conway game A is *winning* when every play $s : x \twoheadrightarrow y$ in the strategy ends on a winning position y, that is, in a position of payoff 0 or $+1$:

$$\text{for all } s \in \sigma, \quad s : x \twoheadrightarrow y \quad \text{implies} \quad \kappa_A(y) \in \{0, +1\}.$$

We define below a category **Payoff** of payoff Conway games whose morphisms from a game A to a game B are the winning strategies on the game $A \multimap B$. In particular, our definition of winning strategy on $A \multimap B$ implies that there exists no winning strategy from a linear game A to an affine game B because the payoff $\kappa_A(\star_A) \multimap \kappa_B(\star_B)$ of the root $\star_{A\multimap B}$ is equal to $+1 \multimap 0 = -1$. In order to define a category, one needs to show that winning strategies do compose.

Proposition 8 *The strategy $\tau \circ \sigma : A \multimap C$ is winning when the two strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$ are winning.*

Proof: We already know that strategies do compose, it just remains to check the winning condition. As it is defined positionally, it suffices to observe by a case analysis that the composite of two winning positions on $\multimap\!\!\bullet$ is winning, in the sense that

$$\begin{array}{ll} \kappa_A(x) \multimap\!\!\bullet\, \kappa_B(y) \in \{0,+1\} & (x \multimap\!\!\bullet\, y : \text{ winning}) \\ \kappa_B(y) \multimap\!\!\bullet\, \kappa_C(z) \in \{0,+1\} & (y \multimap\!\!\bullet\, z : \text{ winning}) \\ \hline\hline \kappa_A(x) \multimap\!\!\bullet\, \kappa_C(z) \in \{0,+1\} & (x \multimap\!\!\bullet\, z : \text{ winning}) \end{array}$$

This works because the definition of the payoff function on $\multimap\!\!\bullet$ comes from the boolean implication $\Rightarrow$, which is itself stable under composition. $\square$

Proposition 9 *The category* **Payoff** *whose objects are payoff Conway games and whose morphisms from A to B are winning strategies on $A \multimap\!\!\bullet\, B$ is symmetric monoidal closed, and in fact $*$-autonomous, with dualizing object the unit game* 1.

Proof: We already know that the category **Conway** is symmetric monoidal closed, and in fact, compact closed. So, in order to establish that the category **Payoff** is symmetric monoidal closed, it is sufficient to check that

$$(\kappa_A(x) \otimes \kappa_B(y)) \multimap\!\!\bullet\, \kappa_C(z) = \kappa_A(x) \multimap\!\!\bullet\, (\kappa_B(y) \multimap\!\!\bullet\, \kappa_C(z))$$

for all positions $x \in V_A$, $y \in V_B$ and $z \in V_C$. This equation is equivalent to the validity of the boolean formula

$$(A \wedge B) \Rightarrow C \equiv A \Rightarrow (B \Rightarrow C)$$

in a three-valued boolean logic. Then, the fact that the category **Payoff** is $*$-autonomous with dualizing object the unit game 1 follows from the observation that the position

$$(x \multimap\!\!\bullet\, \star_1) \multimap\!\!\bullet\, \star_1$$

in the game $(A \multimap\!\!\bullet\, 1) \multimap\!\!\bullet\, 1$ has payoff

$$(\kappa_A(x) \multimap\!\!\bullet\, 0) \multimap\!\!\bullet\, 0 \quad = \quad \kappa_A(x)$$

for every position x of the payoff Conway game A. and the fact that the unique position $\star_1$ $\square$

Note that the category **Payoff** is $*$-autonomous but not compact-closed, because the payoff function distinguishes the tensor product and its dual. On the other hand, the subcategory **Payoff**$^-$ of negative games is related by an adjunction to the category **Payoff** in the same way as the categories **Conway**$^-$ is related to the category **Conway**. In particular, Proposition 6 implies that the category **Payoff**$^-$ is symmetric monoidal closed, with linear implication $A \multimap B$ defined as

$$A \multimap B \quad \stackrel{\text{def}}{=} \quad (A \multimap\!\!\bullet\, B)^-$$

for all negative payoff Conway games A and B.

A dialogue category of games. In order to define an affine modality on our notion of payoff Conway games, it appears necessary to restrict the category $\mathbf{Payoff}^{-}$ to its full subcategory **Games** of negative games whose root is a winning position. Note that every such game is either linear (when the root has payoff $+1$) or affine (when the root has payoff 0). An interesting point is that the category **Games** is still symmetric monoidal but no longer closed because the game $A \multimap B$

- is linear when the game B is linear,
- is affine when the games A and B are both affine,
- is not an object of **Games** when the game A is linear and the game B is affine.

Let $\perp$ define the linear game with two positions $\star$ and **done**, and a unique move $\star \to \mathbf{done}$ played by Opponent, with payoff function

$$\kappa_{\perp}(\star) = +1 \qquad \text{and} \qquad \kappa_{\perp}(\mathbf{done}) = 0.$$

The following property follows immediately from the fact that the game $\neg A$ defined as $A \multimap \perp$ is linear for every game A.

Proposition 10 *The category* **Games** *defines a dialogue category.*

Note that the game $\neg A$ is obtained by reversing the payoff and the role of Player and Opponent in the game A, and by "lifting" the resulting game $A^{*} = A \mathbin{-\!\!\bullet} 1$ with an Opponent move, starting from the root position $\star_{\neg A}$ of payoff $+1$.

5 A game model with resources

We now give an explicit description of the three resource modalities of tensor logic in the dialogue category **Games** constructed in the previous section. We establish the additional properties that the affine and exponential modalities are free, but not the relevant modality.

Affine modality. Recall that a payoff game A is called *affine* when its root is of payoff 0. In that case, the trivial (and unique) strategy t_A from the affine game A to the unit game 1 is winning. The full subcategory of affine games in the category **Games** will be denoted $\mathbf{Games}_w$. We want to show that the embedding functor

$$\mathbf{Games}_w \quad \longrightarrow \quad \mathbf{Games}$$

has a right adjoint, tranporting every game A to the affine game $\underset{w}{!}A$ obtained by assigning the payoff 0 to the root of the game A. A first observation is that the operation $\underset{w}{!}$ defines a functor

$$\underset{w}{!} \quad : \quad \mathbf{Games} \quad \longrightarrow \quad \mathbf{Games}_w$$

which transports every winning strategy $\sigma : A \multimap B$ to itself, seen this time as a winning strategy $\underset{w}{!}\sigma = \sigma$ of the affine game $\underset{w}{!}A \multimap \underset{w}{!}B$. Note that $\underset{w}{!}\sigma$ defines a winning strategy in the game $\underset{w}{!}A \multimap \underset{w}{!}B$ because assigning the null payoff to the roots of the games A and B transports every winning position of $A \multimap B$ to a winning position of $\underset{w}{!}A \multimap \underset{w}{!}B$.

This last point is interesting. It follows from the fact that the root $\star_A \multimap \star_B$ is the only position of the game $A \multimap B$ of the form $a \multimap \star_B$ for a position a of the game A. Indeed, the two games A and B are negative, and hence, Opponent should start the game $A \multimap B$ by playing in the component B. The same reason implies that for every two payoff games A and B,

Lemma 2 *Suppose that the game A is affine. Then, every position $x \multimap y$ of the game $A \multimap B$ (or alternatively of the game $A \multimap !_w B$) satisfies*

$$x \multimap y \text{ is winning in } A \multimap B \quad \text{iff} \quad x \multimap y \text{ is winning in } A \multimap !_w B.$$

Proof: By definition of a winning position as a position with payoff either 0 or +1, and by definition of the payoff function on the games $A \multimap B$ and $A \multimap !_w B$, the property just stated means that

$$\kappa_A(x) \multimap \kappa_B(y) \in \{0, +1\} \quad \text{iff} \quad \kappa_A(x) \multimap \kappa_{!_w B}(y) \in \{0, +1\}.$$

We proceed by case analysis. Suppose that the position y is not the root of B: in that case, the position y has the same polarity in the two games B and $!_w B$, and the property is thus immediate. Suppose that the position y is the root $\star_B$ of the game B: in that case, the position x is necessarily the root $\star_A$ of the game A, by definition of the game $A \multimap B$ as a game where Opponent plays its first move in the component B. So, there remains to compare the payoff of the root of the game $A \multimap B$

$$\kappa_A(\star_A) \multimap \kappa_B(\star_B) \quad = \quad 0 \multimap \kappa_B(\star_B) \quad = \quad \kappa_B(\star_B)$$

which is either 0 or +1, to the payoff of the root of the game $A \multimap !_w B$

$$\kappa_A(\star_A) \multimap \kappa_{!_w B}(\star_{!_w B}) \quad = \quad 0 \multimap 0 \quad = \quad 0$$

and conclude the proof by observing that the root is a winning position in both games. □

Proposition 11 *The functor $!_w$ defines an* affine modality *on the category* **Games**.

Proof: The category of affine games $\mathbf{Games}_w$ is a symmetric monoidal subcategory of **Games**. In particular, the embedding functor from $\mathbf{Games}_w$ to **Games** is symmetric and strong monoidal. The category $\mathbf{Games}_w$ is also an affine category, in the sense that its monoidal unit 1 is terminal. There remains to show that the functor $!_w$ is right adjoint to the embedding functor. To that purpose, one needs to define a natural bijection between $\mathbf{Games}(A, B)$ and $\mathbf{Games}_w(A, !_w B)$ for every affine game A, and every game B. This is precisely the task of Lemma 2, which establishes that a winning strategy of the game $A \multimap B$ is the same thing as a winning strategy of the game $A \multimap !_w B$. The natural bijection is thus defined as the identity. □

It is worth observing that the modality $!_w$ is free, in the sense that the category $\mathbf{Games}_w$ is the category of coalgebras of the comonad $!_w$ on the category **Games**, and at the same time, the slice category $(\mathbf{Games} \downarrow 1)$ over the monoidal unit 1. This last point means that an affine game A may be alternatively defined as a pair consisting of a game A and a morphism $A \longrightarrow 1$ in the category **Games**.

Relevant modality. A diagonal object of a symmetric monoidal category $\mathcal{C}$ is defined as a pair (A, d_A) consisting of an object A and a morphism $d_A : A \longrightarrow A \otimes A$ of the category $\mathcal{C}$ making the Diagrams (7) and (8) commute. In order to construct a relevant modality on the category one starts by defining the category **Diag** as the category whose objects are the diagonal objects (A, d_A) of the category **Games**, and whose morphisms $\sigma : (A, d_A) \longrightarrow (B, d_B)$ are the morphisms $\sigma : A \longrightarrow B$ of the category **Games** making the diagram

$$\begin{tikzcd} A \arrow[r, "\sigma"] \arrow[d, "d_A"'] & B \arrow[d, "d_B"] \\ A \otimes A \arrow[r, "\sigma\otimes\sigma"] & B \otimes B \end{tikzcd}$$

commute. The category **Diag** is symmetric monoidal, with tensor product of two diagonal objects (A, d_A) and (B, d_B) defined as the tensor product $A \otimes B$ of the underlying object, equipped with the morphism

$$A \otimes B \xrightarrow{d_A \otimes d_B} A \otimes A \otimes B \otimes B \xrightarrow{A \otimes symm_{A,B} \otimes B} A \otimes B \otimes A \otimes B\ .$$

Note that the family of morphisms

$$\delta_{(A,d_A)} \quad : \quad (A, d_A) \quad \longrightarrow \quad (A, d_A) \otimes (A, d_A)$$

is natural in (A, d_A) and makes the Diagrams (7) and (8) commute in the category **Diag**. Moreover, the forgetful functor

$$\mathbf{Diag} \quad \longrightarrow \quad \mathbf{Games} \tag{13}$$

is symmetric monoidal. We start the construction of the relevant modality on **Games** by defining a functor in the reverse direction. The idea of the construction is that every game A generates a game ${!}_{c}A$ which may be seen as some kind of infinite tensor product of the game A, defined as follows:

- its positions are the words $w = x_1 \cdots x_k$ whose letters are the positions x_i of the game A different from the root; the intuition is that each letter x_i describes the current position in the i^{th} copy of the game A,

- its root $\star_{{!}_{c}A}$ is the empty word,

- its moves $w \to w'$ are either moves played in one copy :

  $$w_1 \cdot m \cdot w_2 \quad : \quad w_1 \cdot x \cdot w_2 \ \to \ w_1 \cdot y \cdot w_2$$

  where $m : x \to y$ is a move of the game A; or moves where Opponent opens a new copy:

  $$w \cdot m \quad : \quad w \ \to \ w \cdot x$$

  where $m : \star_A \to x$ is an initial move of the game A.

- the polarity of a move $w_1 \cdot m \cdot w_2$ or $w \cdot m$ in the game ${!}_{c}A$ is equal to the polarity $\lambda_A(m)$ of the underlying move m in the game A,

- its payoff function is defined on a position $w = x_1 \cdots x_k$ as

$$\kappa_{\text{¡}A}(w) \quad = \quad \bigotimes_{1 \leq i \leq k} \kappa_A(x_i)$$

according to the payoff table in Figure 13, and on the root as

$$\kappa_{\text{¡}A}(\star_{\text{¡}A}) \quad = \quad \kappa_A(\star_A).$$

Every game $\text{¡}A$ is equipped with a diagonal strategy

$$\delta_A \quad : \quad \text{¡}A \longrightarrow \text{¡}A \otimes \text{¡}A$$

which implements a copycat strategy together with a management of the copy indices. Intuitively, whenever Opponent opens a new copy in the game $\text{¡}A \otimes \text{¡}A$, the strategy δ_A reacts by opening a new copy in the game $\text{¡}A$. Then, the two copies of the game A are linked together until the end of the interaction, in the sense that to every time a move is played by Opponent in one of the two copies, the strategy δ_A reacts by playing the same move in the other copy.

We will define the strategy δ_A in a formal way. To that purpose, we define a function $\langle\rangle$ (called interleaving function) whose task is to translate every play s of the game $\text{¡}A \otimes \text{¡}A$ into a play $\langle s \rangle$ of the same shape in the game $\text{¡}A$. Given a play $s : \star \twoheadrightarrow w_1 \otimes w_2$ and its translation as $\langle s \rangle : \star \twoheadrightarrow w$, we also define a bijective function $[s]$ (called copy function) which associates a copy index in the position w to every copy index in the position w_1 and to every copy index in the position w_2. The two functions $\langle\rangle$ and $[s]$ are defined by induction on the length of the play s in the game $\text{¡}A \otimes \text{¡}A$.

- The empty play of $\text{¡}A \otimes \text{¡}A$ is transported to the empty play of $\text{¡}A$.

- Suppose that the play $s : \star \twoheadrightarrow w_1 \otimes w_2$ of the game $\text{¡}A \otimes \text{¡}A$ is transported to the play $\langle s \rangle : \star \twoheadrightarrow \ w$ of the game $\text{¡}A$. In that case, the play s extended with the move

$$n = (w_1 \cdot m) \otimes w_2 \quad : \quad w_1 \otimes w_2 \to (w_1 \cdot x) \otimes w_2$$

which opens a new copy of A in the left component of the tensor product, is transported into the play $\langle s \rangle$ extended with the move

$$w \cdot m \quad : \quad w \to w \cdot x$$

which opens a new copy of A in the game $\text{¡}A$ with the same move $m : \star \to x$. Moreover, the copy function $[s \cdot n]$ extends the copy function $[s]$ by associating the index of the last opened copy in the position $w_2 \cdot x$ to the index of the last opened copy in the position $w \cdot x$.

- Suppose that the play $s : \star \twoheadrightarrow (w_1 \cdot x \cdot w_2, w)$ of the game $\text{¡}A \otimes \text{¡}A$ is transported into the play $\langle s \rangle : \star \twoheadrightarrow \ w_1' \cdot x \cdot w_2'$ of the game $\text{¡}A$, and that the two copies of A in position x are related by the copy function $[s]$. In that case, the play s extended with the move

$$(w_1 \cdot m \cdot w_2) \otimes w \quad : \quad (w_1 \cdot x \cdot w_2) \otimes w \to (w_1 \cdot y \cdot w_2) \otimes w$$

which plays in the left component of the tensor product is translated into the play $\langle s \rangle$ extended with the move

$$w_1' \cdot m \cdot w_2' \quad : \quad w_1' \cdot x \cdot w_2' \to w_1' \cdot y \cdot w_2'$$

in the game $\underset{c}{!}A$. Moreover, the copy function $[s \cdot n]$ is equal to the function $[s]$.

- We define similarly the interleaving and the copy functions for moves played in the right component of the tensor product.

Using the interleaving function, the strategy δ_A can be defined as

$$\delta_A \quad \overset{\text{def}}{=} \quad \{ s \in \mathrm{Play}^{even}_{\underset{c}{!}A_1 \multimap (\underset{c}{!}A_2 \otimes \underset{c}{!}A_3)} \mid \forall t \preceq^{even} s \ , \ t_{|\underset{c}{!}A_1} = \langle t_{|\underset{c}{!}A_2 \otimes \underset{c}{!}A_3} \rangle \}$$

where the tags 1, 2 and 3 are used to distinguish between the different occurrences of the game A. Note that this strategy is winning because every position $w \multimap (w_1 \otimes w_2)$ it plays has the same payoff $\kappa(w)$ on the left and $\kappa(w_1 \otimes w_2)$ on the right of the linear implication. We also leave the reader check that δ_A satisfies Diagrams (7) and (8) and thus defines a diagonal on the game $\underset{c}{!}A$. This establishes that the pair (A, δ_A) defines an object of the category **Diag**, for every game A. There remains to show that the operation $\underset{c}{!}$ defines a functor

$$\underset{c}{!} \quad : \quad \mathbf{Games} \quad \longrightarrow \quad \mathbf{Diag} \tag{14}$$

which transports every strategy $\sigma : A \multimap B$ into a strategy $\underset{c}{!}\sigma : \underset{c}{!}A \multimap \underset{c}{!}B$ which commutes with the diagonals δ_A and δ_B. This is not particularly difficult, although it should be done with care. An interesting aspect of the construction is that the resulting strategy $\underset{c}{!}\sigma$ plays exactly the positions

$$(x_1 \cdots x_k) \quad \multimap \quad (y_1 \cdots y_k)$$

where the position $x_i \multimap y_i$ is played by the strategy σ, for $1 \leq i \leq k$. The following lemma ensures then that the strategy $\underset{c}{!}\,\sigma$ is winning when the strategy σ is winning, and thus defines a morphism of **Diag**.

Lemma 3 *Suppose that the positions $x_i \multimap y_i$ are winning in the game $A \multimap B$, for $1 \leq i \leq k$. Then, the position $(x_1 \cdots x_k) \multimap (y_1 \cdots y_k)$ is winning in the game $\underset{c}{!}A \multimap \underset{c}{!}B$.*

The question at this point is whether the functor $\underset{c}{!}$ is right adjoint to the embedding functor (13). We show that this is not the case by considering the affine game

$$\mathtt{comm} \quad \overset{\text{def}}{=} \quad \underset{w}{!} \neg\neg\, 1 \quad = \quad \underset{w}{!} \downarrow\uparrow 1.$$

and its winning strategy *run* which reacts to the unique Opponent move by playing the unique Proponent move. The game is affine. It is thus possible to define the strategy

$$d_{\mathtt{comm}} \quad : \quad \mathtt{comm} \quad \xrightarrow{t_{\mathtt{comm}}} \quad 1 \cong 1 \otimes 1 \quad \xrightarrow{run \otimes run} \quad \mathtt{comm} \otimes \mathtt{comm}$$

which defines the diagonal object $(\mathtt{comm}, d_{\mathtt{comm}})$. Now, consider any game A in the category **Games**, and observe that a strategy $\sigma : \mathtt{comm} \to \underset{c}{!}A$ which makes the diagram

$$\begin{tikzcd}
\mathtt{comm} \arrow[rr, "\sigma"] \arrow[d, "t_A"'] & & \underset{c}{!}A \arrow[dd, "\delta_A"] \\
1 \cong 1 \otimes 1 \arrow[d, "run \otimes run"'] & & \\
\mathtt{comm} \otimes \mathtt{comm} \arrow[rr, "\sigma \otimes \sigma"] & & \underset{c}{!}A \otimes \underset{c}{!}A
\end{tikzcd}$$

commute factors through the game 1, and thus, does not play any move in the game `comm`. We leave the reader establish that there is a one-to-one relationship between these strategies and the strategies of the game A:

$$\mathbf{Diag}((\texttt{comm}, d_{\texttt{comm}}), !_c A) \quad \cong \quad \mathbf{Games}(1, A).$$

On the other hand, if the functor $!_c$ was right adjoint to the forgetful functor (13), there would be a bijection between the sets

$$\mathbf{Diag}((\texttt{comm}, d_{\texttt{comm}}), !_c A) \quad \cong \quad \mathbf{Games}(\texttt{comm}, A).$$

This establishes that the functor $!_c$ is not right adjoint to the forgetful functor, since there exists no such bijection between $\mathbf{Games}(\texttt{comm}, A)$ and $\mathbf{Games}(1, A)$: simply consider the particular case when $A = \texttt{comm}$.

This shows that the relevant modality $!_c$ we are constructing is not *free* in the category **Games**. This mainly comes from a lack of compatibility between the duplication and the weakening, which enables any affine game A (with a strategy) to define a diagonal game (A, d) whose morphism $d : A \longrightarrow A \otimes A$ does not implement any duplication mechanism. We will see in the next paragraph that this kind of fake duplication is rejected when one considers commutative comonoids instead of diagonal objects.

The category $\mathbf{Games}_c$ is defined as the full subcategory of **Diag** consisting of all the diagonal objects isomorphic in **Diag** to a tensor product of the form:

$$!_c A_1 \ \otimes \ \cdots \ \otimes \ !_c A_k.$$

By construction, the category $\mathbf{Games}_c$ is symmetric monoidal, and the composite functor

$$\mathbf{Games}_c \quad \longrightarrow \quad \mathbf{Diag} \quad \longrightarrow \quad \mathbf{Games}$$

is symmetric monoidal. It appears moreover that the functor is left adjoint to the functor $!_c$ seen this time as a functor from **Games** to $\mathbf{Games}_c$. This enables us to conclude that

Proposition 12 *The functor $!_c$ defines a* relevant modality *on the category* **Games**.

Exponential modality. From now on, $\mathbf{Games}_e$ will denote the category whose objects are the commutative comonoids of the category **Games**, and whose morphisms are the comonoid morphisms between them. The exponential modality

$$!_e \ : \ \mathbf{Games} \quad \longrightarrow \quad \mathbf{Games}_e$$

is obtained by applying successively the affine and the relevant modality, in any order:

$$!_e \quad \overset{\text{def}}{=} \quad !_c\, !_w \quad = \quad !_w\, !_c.$$

Observe in particular that the commutation of $!_w$ and $!_c$ induces two distributivity laws in the sense of Beck [6]:

$$!_w\, !_c \ \longrightarrow \ !_c\, !_w \qquad \text{and} \qquad !_c\, !_w \ \longrightarrow \ !_w\, !_c.$$

We establish that

Proposition 13 *The functor $!_e$ defines an* exponential modality *on* **Games**.

Proof: The distributivity law $!_w !_c \to !_c !_w$ implies that the comonad $!_c$ extends to a comonad on the category $\mathbf{Games}_w$ of coalgebras of the comonad $!_w$. It appears then that this comonad is induced by a symmetric monoidal adjunction

$$\begin{tikzcd} \mathbf{Games}_e \arrow[rr, bend left, "U"] \arrow[rr, phantom, "\perp"] & & \mathbf{Games}_w \arrow[ll, bend left, "!_c"] \end{tikzcd}$$

whose left adjoint is the forgetful functor V from the category $\mathbf{Games}_e$ to the category $\mathbf{Games}_w$, and whose right adjoint is the functor $!_c$ defined in Equation (14), restricted to the full subcategory $\mathbf{Games}_w$ of affine games. We conclude by observing that the two symmetric monoidal adjunctions compose

$$\begin{tikzcd} \mathbf{Games}_e \arrow[rr, bend left, "V"] \arrow[rr, phantom, "\perp"] & & \mathbf{Games}_w \arrow[ll, bend left, "!_c"] \arrow[rr, bend left, "U"] \arrow[rr, phantom, "\perp"] & & \mathbf{Games} \arrow[ll, bend left, "!_w"] \end{tikzcd}$$

and define a symmetric monoidal adjunction

$$\begin{tikzcd} \mathbf{Games}_e \arrow[rr, bend left, "U \circ V"] \arrow[rr, phantom, "\perp"] & & \mathbf{Games}. \arrow[ll, bend left, "!_e"] \end{tikzcd}$$

This adjunction means that the exponential modality $!_e$ is free, in the sense that it computes the free commutative comonoid $!_e A$ generated by an object A in the category **Games**. A pleasant way to establish this fact is to apply the general formula for computing the free exponential modalities, which the interested reader will find in [38]. $\Box$

Products. The category **Games** does not have finite coproducts and thus does not interpret tensor logic with additives. This provides us with a nice opportunity to apply the family construction described in Section 3, in order to construct a model of tensor logic with additives and exponentials.

Given a family $(A_i)_{i \in I}$ of objects of the category **Games** indexed by a set I, the product $\&_{i \in I} A_i$ in the category **Games** is defined as follows:

- its underlying graph is obtained by taking the disjoint union of the graphs underlying each game A_i, and by merging their root,
- the polarity of moves is directly inherited from the polarity of the moves in A_i;
- the payoff function is inherited from the payoff function of each A_i, except for the root. The root has payoff $+1$ when all the roots of the A_i have payoff $+1$, and had payoff 0 when one of the roots of the A_i has payoff 0.

In other words, the game $\&_{i \in I} A_i$ is linear when all the games A_i are linear, and affine otherwise. The i^{th} projection is provided by the obvious copycat strategy between the game A_i and the i^{th} component of the game $\&_{i \in I} A_i$.

The category $Fam(\mathbf{Games})$. Here, we deduce from Theorem 2 at the end of §3 that the category $Fam(\mathbf{Games})$ is a dialogue category equipped with an affine, a relevant and an exponential modality. We conclude that:

Proposition 14 *The category $Fam(\mathbf{Games})$ is a model of full propositional tensor logic (multiplicative, additive, affine, relevant, exponential).*

Note that a family $(A_i)_{i \in I}$ may be seen alternatively as a positive payoff game, whose Player moves from the root are the indices i, leading to the negative game A_i, and whose root has payoff 0. In this way, the category $Fam(\mathbf{Games})$ may be seen as a subcategory of the category $\mathbf{Payoff}^+$ of positive games, whose morphisms $A \longrightarrow B$ are "transversal" strategies, which always react to an opening move $i \in I$ played in the positive game $A = (A_i)_{i \in I}$ by playing an opening move $j \in J$ in the positive game $B = (B_j)_{j \in J}$.

A remark on multi-bracketed Conway games. The category of multi-bracketed Conway games introduced by the two authors in [36] provides another model of tensor logic. The category **Games** is a much simpler model of the logic, but it is based on the category **Payoff** which is $*$-autonomous, instead of compact closed. The model based on multi-bracketing is precisely designed to preserve the compact closed structure of the original category **Conway** of Conway games. From this, one extracts a canonical trace operator, which then plays an important role in the game-theoretic description of the memory cell of a functional language with general references [37]. So, summarized in a few words, the category **Games** describes a simple model of tensor logic, based on Conway games equipped with a payoff function, whereas the category of multi-bracketed Conway games introduced in [36] describes a more complicated model of tensor logic, but equipped this time with a trace operator.

6 Conclusion

In this paper, we integrate resource modalities in game semantics, a task which has been considered difficult to accomplish in the past. The task requires indeed to put many ideas together, and to reunderstand the topography of the field. In particular, linear logic is refined here into tensor logic, where the involutive negation of linear logic is replaced by a more general (and non involutive) notion of tensorial negation. In that way, it becomes possible to keep the best of linear logic: its proof theory, its resource modalities, etc. and to work on games and continuations instead. Moreover, it appears that linear logic coincides with tensor logic with the additional axiom that the continuation monad is commutative. In that sense, tensor logic is more primitive than linear logic, in the same way that groups are more primitive than abelian groups. So, this work reunifies the fields of linear logic and game semantics, and opens the nice research problem of understanding how the principles and constructions of linear logic should be refined (and extended) in order to apply to game semantics and tensor logic.

References

[1] S. Abramsky, Sequentiality vs. concurrency in games and logic, Mathematical Structures in Computer Science 13 (2003) 531–565.

[2] S. Abramsky, R. Jagadeesan, Games and full completeness for multiplicative linear logic, Journal of Symbolic Logic 59 (2) (1994) 543–574.

[3] S. Abramsky, R. Jagadeesan, P. Malacaria, Full abstraction for PCF, Information and Computation 163 (2) (2000) 409–470.

[4] S. Abramsky, G. McCusker, Call-by-value games, in: M. Nielsen, W. Thomas (eds.), 6th Annual Conference of the European Association for Computer Science Logic, vol. 1414 of Lecture Notes in Computer Science, Springer, 1998.

[5] J.-M. Andréoli, Logic programming with focusing proofs in linear logic, Journal of Logic and Computation 2 (3).

[6] J. Beck, Distributive laws, in: Seminar on Triples and Categorical Homology Theory, vol. 80 of Lecture Notes in Mathematics, 1969.

[7] N. Benton, A mixed linear and non-linear logic: Proofs, terms and models, in: Third Annual Conference of the European Association for Computer Science Logic, vol. 933 of LNCS, Springer-Verlag, Poland, 1995.

[8] G. Berry, P.-L. Curien, Sequential algorithms on concrete data structures, Theoretical Computer Science 20 (1982) 265–321.

[9] A. Blass, Degrees of indeterminacy of games, Fundations of Mathematics 77 (1972) 151–166.

[10] A. Blass, A games semantics for linear logic, Annals of Pure and Applied Logic 56 (1992) 183–220.

[11] R. Cockett, R. Seely, Polarized category theory, modules, and game semantics, Theory and Applications of Categories 18 (2) (2007) 4–101.

[12] T. Ehrhard, Finiteness spaces, Mathematical Structures in Computer Science 15 (04) (2005) 615–646.

[13] J.-Y. Girard, Linear logic, Theoretical Computer Science 50 (1987) 1–102.

[14] J.-Y. Girard, A new constructive logic: Classical logic, Mathematical Structures in Computer Science 1 (3) (1991) 255–296.

[15] R. Harmer, Games and full abstraction for nondeterministic languages, Ph.D. thesis, University of London (2000).

[16] M. Hasegawa, Private communication (July 2005).

[17] M. Hofmann, T. Streicher, Completeness of continuation models for $\lambda\mu$-calculus, Information and Computation 179 (2) (2002) 332–355.

[18] M. Hyland, L. Ong, On full abstraction for PCF: I, II and III, Information and Computation 163 (2) (2000) 285–408.

[19] M. Hyland, J. Power, Pseudo-commutative monads and pseudo-closed 2-categories, Journal of Pure and Applied Algebra 175 (1-3) (2002) 141–185.

[20] B. Jacobs, Semantics of weakening and contraction., Annals of Pure and Applied Logic 69 (1) (1994) 73–106.

[21] A. Joyal, Remarques sur la théorie des jeux à deux personnes, Gazette des Sciences Mathématiques du Québec 1 (4) (1977) 46–52, english version by Robin Houston available.

[22] M. Kelly, M. Laplaza, Coherence for compact closed categories, Journal of Pure and Applied Algebra 19 (1980) 193–213.

[23] A. Kock, On double dualization monads, Mathematica Scandinavica 27 (1970) 151–165.

[24] O. Laurent, Etude de la polarisation en logique, Ph.D. thesis, Université Aix-Marseille II (2002).

[25] O. Laurent, Polarized games (extended abstract), in: 17th IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, Copenhagen, 2002.

[26] O. Laurent, L. Regnier, About translations of classical logic into polarized linear logic, in: 18th IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, Ottawa, 2003.

[27] P. Lorenzen, Ein dialogisches Konstruktivitatskriterium, Infinitistic Methods (1961) 193–200.

[28] P. Lorenzen, K. Lorenz, Dialogische Logik, Wissenschaftliche Buchgesellschaft, 1978.

[29] G. McCusker, Games and full abstraction for a functional metalanguage with recursive types, Ph.D. thesis, University of London (1996).

[30] P.-A. Melliès, Asynchronous games 2: the true concurrency of innocence, in: P. Gardner, N. Yoshida (eds.), Proceedings of the 15th International Conference on Concurrency Theory (CONCUR 2004), No. 3170 in LNCS, Springer Verlag, 2004.

[31] P.-A. Melliès, Asynchronous games 3: an innocent model of linear logic, Electronic Notes in Theoretical Computer Science 122 (2005) 171–192.

[32] P.-A. Melliès, Asynchronous games 4: a fully complete model of propositional linear logic, in: 20th IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, Chicago, 2005.

[33] P.-A. Melliès, Functorial boxes in string diagrams, in: 15th Annual Conference of the European Association for Computer Science Logic, vol. 4207 of Lecture Notes in Computer Science, Springer, 2006.

[34] P.-A. Melliès, Categorical semantics of linear logic, in: Interactive models of computation and program behaviour, Panoramas et Synthèses, Société Mathématique de France, 2009.

[35] P.-A. Melliès, P. Selinger, Games are continuation models!, talk at Full Completeness and Full Abstraction, Satellite workshop of LICS 2001.

[36] P.-A. Melliès, N. Tabareau, Resource modalities in game semantics, in: 22th IEEE Symposium on Logic in Computer Science, IEEE Computer Society, Warsaw, 2007.

[37] P.-A. Melliès, N. Tabareau, An algebraic account of reference in game semantics, in: 25th conference on Mathematical Foundations of Programme Semantics, ENTCS, 2009.

[38] P.-A. Melliès, N. Tabareau, C. Tasson, An explicit formula for the free exponential modality of linear logic, in: 36th International Colloquium on Automata, Languages and Programming, July 2009, Rhodes, Greece, 2009.

[39] E. Moggi, Notions of computation and monads, Information and Computation 93 (1991) 55–92.

[40] J. Power, E. Robinson, Premonoidal categories and notions of computation, Mathematical Structures in Computer Science 7 (1997) 453–468.

[41] P. Selinger, Control categories and duality: on the categorical semantics of the $\lambda\mu$-calculus, Mathematical Structures in Computer Science 11 (2) (2001) 207–260.

[42] N. Tabareau, Modalités de ressources et contrôle en logique tensorielle, Ph.D. thesis, Université Paris 7 – Denis Diderot (2008).

[43] H. Thielecke, Continuation semantics and self-adjointness, Electronic Notes in Theoretical Computer Science 6 (1997) 348–364.